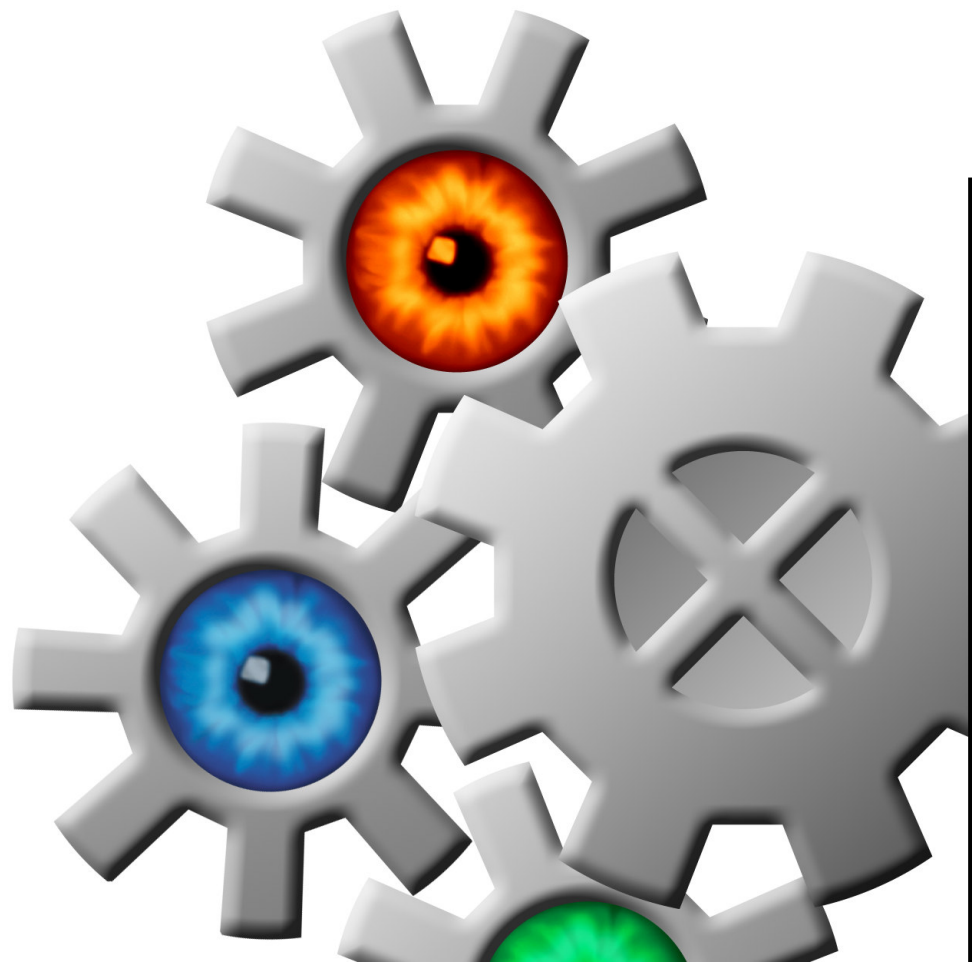


eEye® Digital Security

Pilot Guide for Blink



Blink V2.0

V2.0 is an extension of the Blink V1.6X functionality. Everything in this guide is applicable to both code sets.

Notes on V2.0:

- Spyware quarantining is automatic. If you enable quarantining in the Options->Anti-spyware tab you will automatically quarantine the discovered files. To restore them you may right-click on them in the Event Log.
- Anti-Identity Theft rules are enabled and disabled by from the grid of rules below the Home Page icon.
- Application Protection, which is eEye's advanced buffer overflow protection is enabled and disabled by selecting the Enable/Disable button below the Home Page icon. Detailed, advanced setting may be done in a configuration file described in Appendix A below.

Contributors

The Pilot Guide is based on contributions from many eEye team members, from our Sales Engineers who work daily with customers to the R+D driver developers. You will find basic installation to detailed debugging steps outlined.

Guide Objective

This guide is intended to be a supplemental reference to Blink training and the User and Installation Guides as provided by eEye. The intended audiences are eEye customers, employees, partners and professional consultants; or anyone who is performing a Blink pilot and deployment.

The objective of this guide is to help you configure and deploy a Blink installation and help you explore or demonstrate the features of Blink with a minimum amount of required reading. This guide is designed to show you how Blink has the following features and benefits :

- **Firewalling** - Blink has complete TCP/IP firewalling capabilities
- **Application Firewalling** – Blink can monitor or define application usage policies
- **Spyware and Malware Protection** – Blink can scan disk and memory, and quarantine bad executables
- **Identity Theft Protection** – Blink blocks suspicious Web pages
- **Buffer Overflow Protection** – Blink blocks the root actions of BO activity using the most sophisticated memory protection available today.
- **IPS Using Signature Analysis** – Blink has “out of the box” signatures and allows users to create signatures based on protocols, not just general IP payload content

-
- **IPS Using Data Analysis** - Blink has intrusion protection that protects the kernel before the attack gets into the Kernel.
 - Unlike CSA, Sana or McAfee which warn you after the damage has been done, Blink catches the attack before it is in kernel memory.
 - **Enterprise Support** - Agent deployment and policy updates are centrally managed by the Security Console.
 - **Vulnerability Assessment** - Blink assesses host vulnerability based on the Retina Vulnerability Scanner.

Table of Contents

Table of Contents

Blink Enterprise Installation	5
Installation Summary	5
Architectural Overview of Blink.....	6
Installation and Product Overview	6
System Requirements	8
Additional Notes	8
Overview of Installing the Blink Workstation Agent	10
<i>Note that this is also the first step to creating a Security Console.</i>	10
Installing the eEye Security Console	13
Demonstrating the Security Console.....	15
Overview.....	15
Network Devices View.....	17
Network Tasks.....	19
Central Policy View	21
Overview	21
Creating a Central Policy	21
Package Creation and Deployment.....	23
Once a Policy is defined you need to deploy it with the Blink software to a host that will become a Blink Agent. To perform this, use the "Tools View".....	23
Deploying a Blink Package to Target Hosts	24
Overview	24
Demonstrate the Package/Agent that has been Deployed	25
Demonstrate Deployment of a New Policy.....	26
Conclusion.....	28
Blink Installation Process Template	29
Overview.....	29
Phase One - Planning.....	29
Phase Two - Build Objectives & Agreement.....	30
Phase Three - Proof of Concept, Technology Evaluation.....	30
Phase Four - Review of Findings.....	30
Scope of Deployment.....	31
Installation	31
Deployment.....	31
Management.....	32
Additional Client Specified Criteria	33
Blink Q+A and Tips	33
1. I want to update the Blink product asynchronously from a command line.	33
2. How do I get a quick hands-on review of the Blink features.	33
3. I want to know how the hosts are being discovered :.....	34
4. How do I remotely debug Blink?.....	34
5. How do I deploy just the Blink standalone application?	34
6. I don't want updates coming directly from eEye. How do I distribute them from my server? 34	
7. How do I check if all packages arrived at the target host, and their history of deployment? 34	
8. How do I enable or check IPC\$ file sharing from my server?	34
9. How can I add some of our in-house scripts to Blink?	37
10. Can Blink be used for quarantining?	37
11. What will Blink do in our environment?.....	37
12. How do I configure Blink for Exchange Server?	37
13. How do I configure Blink for a Domain Controller?	38

Table of Contents

14.	How do I know what Blink is blocking?	39
15.	How does Blink stop Shareware?	39
16.	How can I include scripts in Blink?	39
17.	Can Blink secure my wireless laptops?.....	40
18.	How does Blink help me comply with Sarbox 404?	40
19.	Does Blink run with NetWare?	41
20.	Does Blink run with Windows Terminal Server and Citrix?	41
21.	What is Blink Application Protection?	42
22.	What is Blink API Protection?.....	42
23.	How does Blink work with EAP, EAPOL, 802.1X and RADIUS ?	43
24.	How does Blink interoperate with TCB ?.....	44
25.	How can I use Blink stop Spyware and Malware?	44
26.	How do I allow applications to run that look like Spyware or Malware?	45
27.	How does Blink stop Phishing and Identity Theft?	46
28.	How much memory does Blink consume?.....	46
29.	How much CPU does Blink consume?	46
30.	How do I use a proxy server for Blink licensing?	46
31.	How does Blink stay up to date?	46
32.	Where are we today, and what is the roadmap?	46
33.	How should I update my agents after they have been deployed?	47
34.	What processes does Blink run?.....	47
35.	Do I need .NET?	48
36.	Blink filters on FTP appear to not work	48
37.	Troubleshooting: How do I troubleshoot the Blink installation?	48
38.	Troubleshooting: How do I troubleshoot the Console installation?	48
39.	Troubleshooting: Blink Security Console will not start:	48
40.	Troubleshooting: Blink agent debugging:	49
41.	Troubleshooting: Blink protocol processing debugging:.....	50
42.	Troubleshooting: Blink deployment debugging:	50
43.	Troubleshooting: Blink product update debugging:	50
44.	Troubleshooting: Blink communications debugging:	50
45.	Troubleshooting: Blink Security Console debugging:	50
46.	Troubleshooting: Blink does not seem to filter my traffic:	51
47.	Troubleshooting: Seeing error, "Could not update Product Names":	51
48.	Troubleshooting: communication between the agents and the Security Console:	52
49.	Troubleshooting: Blink Wireless debugging:	52
50.	Troubleshooting: Blink Script Engine debugging:	52
51.	Troubleshooting: "Could not update Product Names":	53
52.	Troubleshooting: SyncIt – Blink product updating:.....	53
53.	Troubleshooting: Blink asked for the license key twice because it's expired:.....	53
54.	Troubleshooting: Blink I/O "The root element is missing":	53
55.	Troubleshooting: Can't get past the Admin credentials prompts:.....	54
56.	Troubleshooting: How to enable Application Protection in Blink 1.X Versions:	54
57.	Troubleshooting: How remove Blink and Security Console Manually:	54
58.	Troubleshooting: Blink to Security Console Communication Requirement:.....	54
59.	Troubleshooting: Security Console Discovery View:.....	55
Appendix A.....		57
Application Protection Configuration		57

Blink Enterprise Installation

Installation Summary

In a pilot installation it is recommended that you use the standard Security Console configuration where the Security Console manages Blink workstation configurations and deployment. However an alternative to using the Security Console to deploy Blink agents, is to deploy Blink from a central software distribution system to client workstations and use a Security Console to manage the organization's security policies. The Blink "Third Party Deployment Tool", enables users to use IBM's Tivoli, SMS, CA's Unicenter, and scripts to automatically deploy Blink throughout an enterprise or a department.

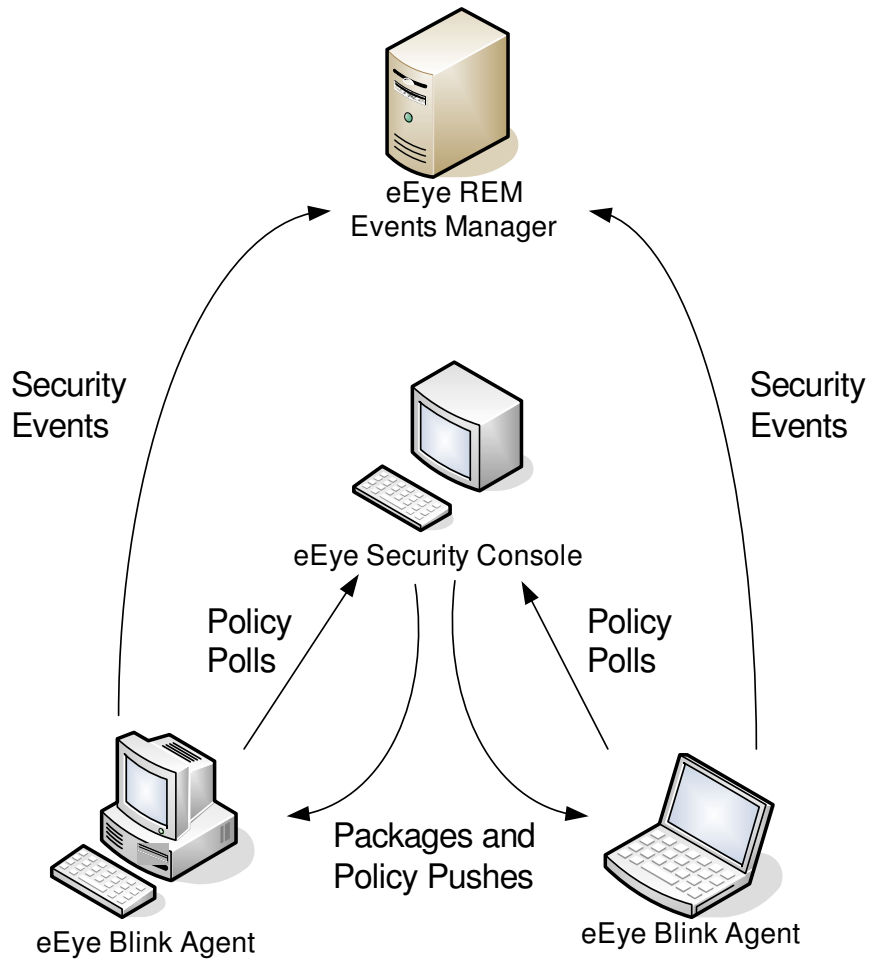
A complete (final) Blink pilot should include REM, with a Security Console managing Blink workstation agents, as mentioned above. This should be performed with at least three systems so that each component's role will be clearly understood:

- REM Manager – aggregates events from eEye products
- Security Console – manages the configuration of Blink agents
- Blink Workstation – workstations that are protected by Blink

Blink workstation agents are managed by the Security Console. The SC can deploy agents to target hosts that need to be protected by Blink. Additionally it manages the policy deployment and updates.

Blink workstations send events to REM which consolidates and analyzes them. REM then uses that digested information to produces reports. This architecture is depicted below.

Architectural Overview of Blink



- Firewall
- Application Monitoring
- Protocol Analysis
- Signature Detection
- Vulnerability Audit

Installation and Product Overview

Install the Blink components in the following order: (note the manual explains these steps in detail)

1. Designate a **Security Console**, which should be a server class system (the system requirements are below) . This host will be the central Blink manager that you will

use to deploy Blink agents and policies from. This host requires both the workstation Blink and the Security Console to be installed.

2. **Install the Blink workstation agent** as described in the Blink Installation Guide or as described below. (This server should be the one that you are planning to make into a Security Console, i.e. you install Blink first then the Security Console software on this host.)
3. Next **install the Blink Security Console** on the **same server** as the agent described above. To Deploy a Blink Package the Security Console uses a copy of the Blink agent; thus both the agent and the console software are loaded into the Security Console.
4. After starting the Security Console select a target host/workstation from the Discovery View and **create a Blink policy**. This creates and names a set of rules.
5. From the Security Console select a target host/workstation from the Discovery View and **“deploy” a Blink package to the workstation**. This installs the agent on the target host.
6. Optionally deploy new policies and packages to more target workstations that you want protected in the pilot.
7. Finally if you want Blink to access the REM Event Server, install REM on another host and configure Blink agents to send events to this host.

System Requirements

To install Blink you must meet at least the minimum system requirements described as follows. For better system performance, eEye recommends the system upgrades described below. Minimum System Requirements

1) Security Console

The following describes the minimum system required to install the **eEye Security Console**:

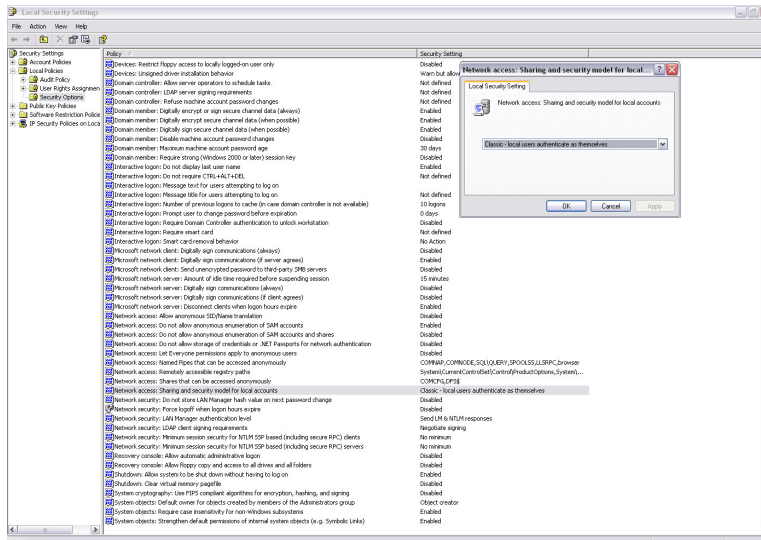
- OS Workstation: Windows NT 4 (SP6), Windows 2000 (SP3), or Windows XP
- OS Server: Windows NT Server (SP6), Windows 2000 Server, Windows 2000 Advanced Server, or Windows Server 2003
- 233 MHz or higher Intel Pentium II or compatible processor
- 256 MB RAM
- 50 MB hard-disk space required for installation
- .Net Framework 1.1 – Note that this is for the console only

2) Agent

Blink agent requirements are similar to those of the Security Console, listed above, except that .NET is not required and you may lower the memory limit down to 128 MB RAM.

Additional Notes

- If workstations are not in a Windows domain, the Local Access Policies for the host must have Network Access set to **Classic Mode** versus Guest Mode.

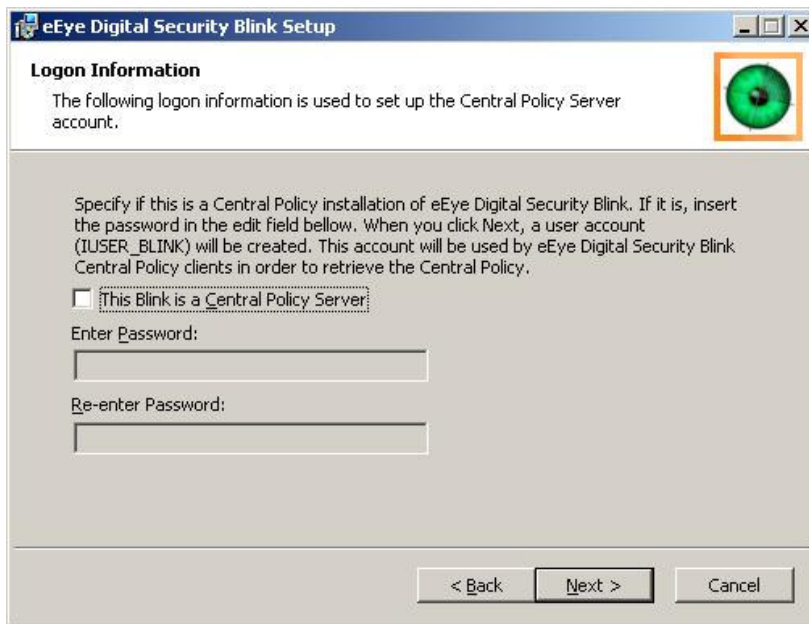


Overview of Installing the Blink Workstation Agent

Note that this is also the first step to creating a Security Console.

Note that the User Manual has complete installation instructions. This section streamlines the process down to the key points (to get started quickly).

1. Begin by running the Blink installation program on the host that you want to have as the Security Console or a standalone Blink instance.
2. The first decision you must make is when the Logon Information window appears.



If you are planning to make this host a Security Console, (a console that will manage other Blink agents) specify a password for the user IUSER_BLINK. (This user will automatically be created and used by the Blink Agent to Security Console communication when the workstation agent polls for a new policy.) You will never see or use this password, it will be used internally by the Blink agents and the Security Console to communicate.

3. If possible, do not change the Destination Folder (figure below) since this will only add to support requirements for a pilot. (It is just simpler to use the default unless you must redefine the installation path.)

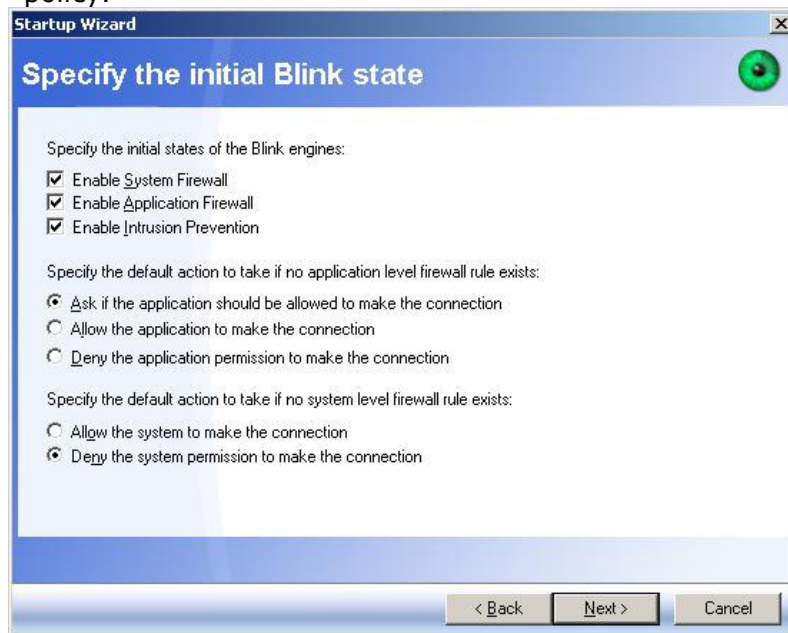


4. **A system restart is necessary** so that installation changes take affect.

At this point the system that the agent is being installed on has rebooted.....

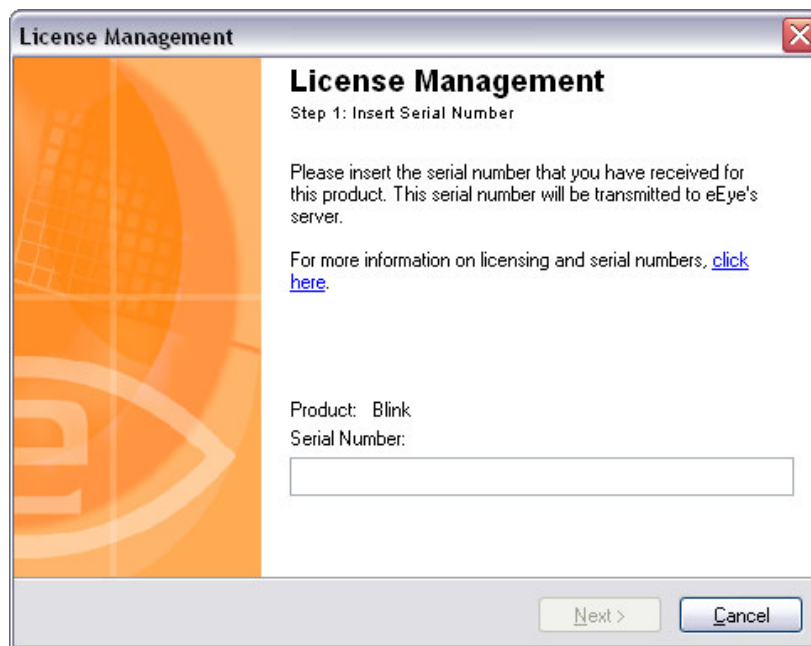
5. After restarting the Blink Startup Wizard comes up. One of the key dialogs is the "Initial Blink State" which is where you define the initial firewalling, application, IPS signature and data analysis rule settings. The choices for the mode that Blink will run in are:
 - a. Interactive - which means that users will be prompted whether to allow traffic to pass
 - b. Alert - display the "toaster" alerts when a rule triggers
 - c. Silent - only log events if logging is enabled for the rules that trigger
6. Next you will choose what layers of defense to enable:
 - a. Firewall
 - b. Application
 - c. Intrusion Prevention - signatures and protocol analysis
 - d. What to do when an application accesses the network

- e. What to do with frames that do not trigger a rule. For example you might write rules to only allow traffic on ports X, Y and Z. When a frame on another port arrives the default action can be set to "Deny", which would then enforce this policy.



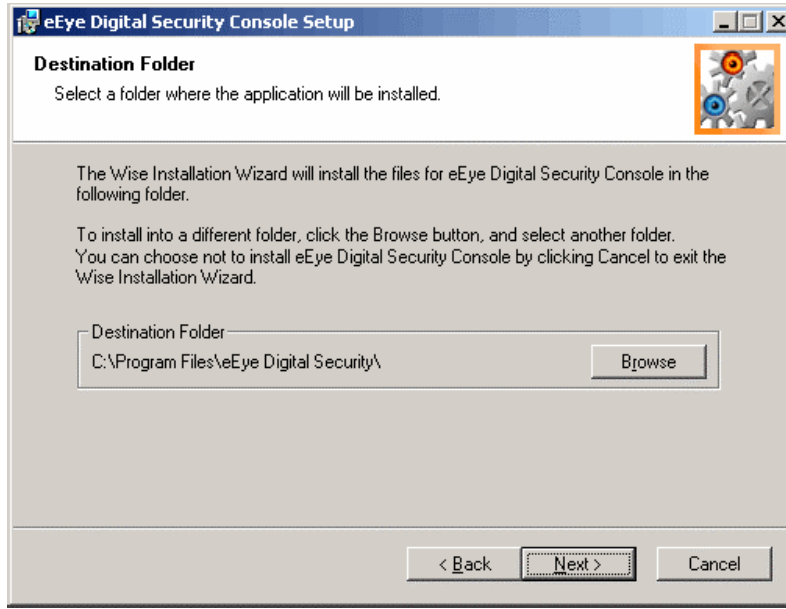
7. When the wizard dialog box asks for the **central policy location**, {"Use central policy"}) you will see the dialog prompt.
- For a Security Console Installation: You should leave this blank. The policy for this host (the console) is implicitly the policy this Security Console host will be configured with.
 - For a Blink agent installation:
 - a. Point this to the Security Console or another server that is acting as a Central Policy server. (This can also be an HTTP, HTTPS or FTP server. For an initial Blink installation we highly recommend that you use the Security Console.)
 - a. Enter the URI of the Central Policy server. The format will be as follows:
eom://hostname/cp_filename.xml
 1. For example: eom://mySecurityConsole/policyA.xml
 - b. Enter the password for IUSER_BLINK in the **Password** field.
 - b. Or you can leave this blank if you are performing a standalone Blink agent installation with no Security Console.
8. The next dialog lets you define who will be Blink Administrators
9. The final step is to enter your product key. You can get this by logging into your customer account page at eEye. eEye will provide you with a login and password so that

you can access key, product and documentation for Blink. **This key manages a pool of licenses. You will not be required to manage keys for each Blink Agent.**



Installing the eEye Security Console

1. After installing the Blink agent in this host, run the blink setup executable (select the file in a Windows file explorer) .
2. Select or run ConsoleSetup.exe.
3. Accept the License Agreement etc...
4. If possible, do not change the Destination Folder (figure below) for a pilot since this will only add to support requirements.



5. Start the Security Console

Demonstrating the Security Console

This section is intended to give an overview of Blink's features. It is not a comprehensive User Guide.

Overview

The Security Console displays the following shortcut bars on the left side of the window:

- **View Network Devices** - this is the network tree view as discovered by ARP, NetBIOS and Active Directory
- **Blink Central Policy** - The Security Console provides configuration functions for the administration of Blink agents. A Central Policy is comprised of Blink settings for instance:
 - Blink Firewall Rules
 - Application Rules
 - IPS rules
- **Blink** – The features of each Policy that can be pushed to workstations.
 - This is virtually the same as the standalone Blink UI for each workstation; thereby a demo of this will cover the workstation functionality
- **Tools** – Important management features

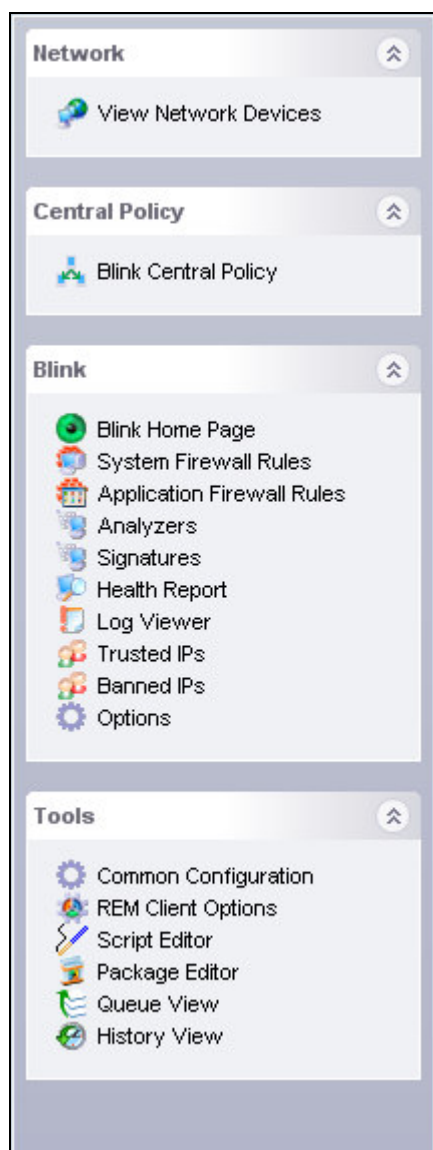
Areas to Explore

A suggested overview of the order of areas to explore are:

- The **Discovery Process**
 - Blink uses ARP, NETBIOS and Active Directory
 - Users may also use lists of hosts to import into the Discovery View.
 - Right click any area in the tree of discovered hosts and in the options list you can select the discovery method options
- **Policy creation** – create a set of rules for Blink Agents
 - Policies can be defined by department, location and environment
 - As examples of the types of Policies you can define:
 - You can have a "Quarantine Policy", an "Enterprise Connected Policy" and an "Off-site Policy"
 - Policies are defined under the category, "Blink Central Policy"
 - Create a new policy by pushing the "New" button
 - Discuss or review the features under the tabs:
 - Firewall Rules – set port and IP address filters
 - Application Rules – filter applications
 - Signatures – "out-of-the-box" and user defined patterns
 - Analyzers – "the secret sauce" from eEye's R+D research team
 - Trusted and Untrusted IP addresses – used to override all other rules

- **Package creation and deployment** – create a Blink software package that includes an initial Policy for your Blink Agents.
 - In the Security Console shortcut bars on the left side of the window choose Package Editor
 - Choose Create
 - Give your package a name, description...
 - Review the package options. Most importantly:
 - Policy Updating – the poll interval that the agent will make for changes to its policy
 - Deploy the package from the Discovery View by right clicking on a host

- **Blink Components** - From Blink home page for each target – (right click on a host and choose “connect”)
 - Review the rules in each target host
 - Discuss the Retina Vulnerability Scanning
 - Start a scan
 - Review the report



Network Devices View

The **Network Devices View** displays a list of network hosts (workstations, routers, servers, printers, etc...) that your Security Console has network awareness of (the console queries ARP, AD and NetBIOS for a list of known hosts/devices).

The following columns of information about the device will be displayed if available:

- **Name** – the NetBIOS name of the target device
- **IP** – the IP of the target device
- Products Installed on the target device
- **Class A** – the address of the target device
- **Class B**
- **Class C**

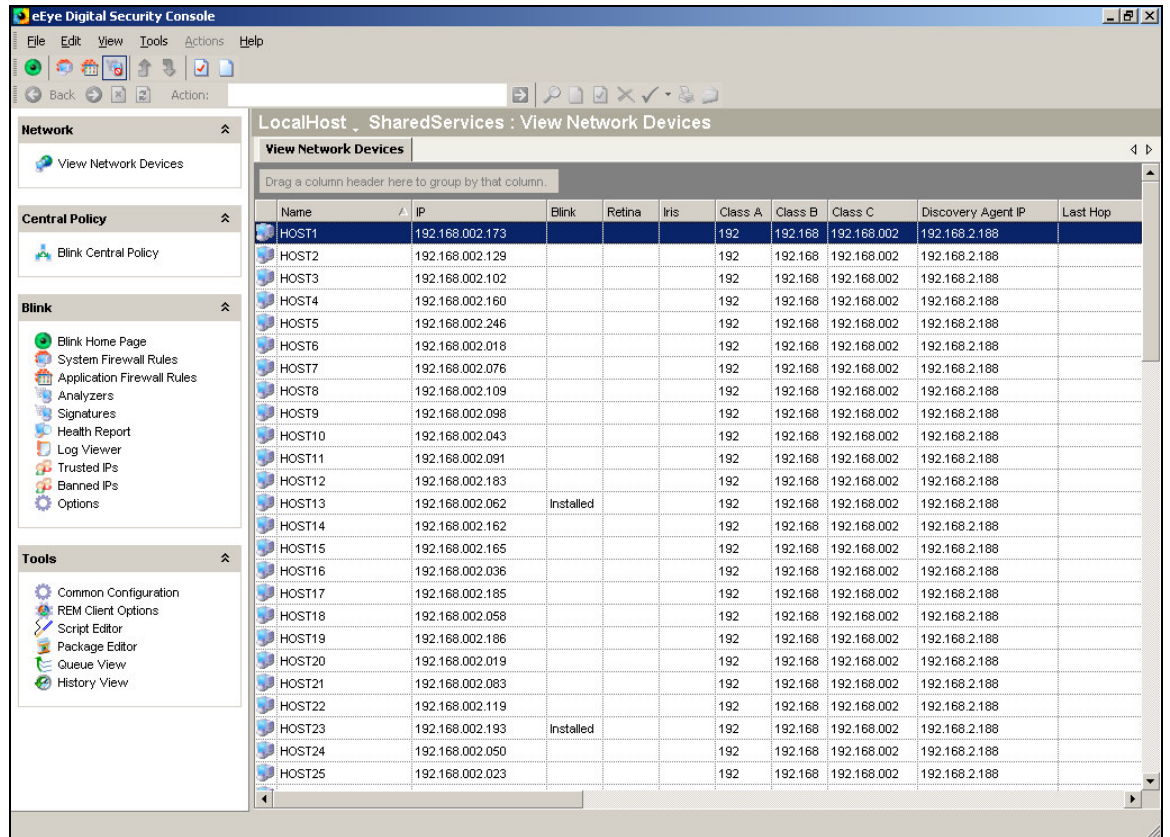
- **Discovery Agent IP** – the address of the machine that discovered this device
- **Last Hop** – the IP of the last hop in the traceroute (if there are multiple hops)
- **DNS Name** – the DNS name of the device, if available
- **Type** – the type of device located at this address
- **Categories**
- **Domain Name** – the name of the parent domain
- **OS Name** – type of operating system (OS)
- **OS Version** – the manufacturer’s version number of the OS
- **OS ServicePack** – the patch level of the device, if available
- **MAC Address** – the address of the NIC that this record refers to
- **MAC Alias**
- **Description**
- **Network Name** – the type of network OS
- **Last Modified**

Note that this Network Devices View is not a systems management tool and is primarily used to determine devices that should have eEye security products deployed.

You can group nodes by their Class C address, OS Version etc... to help organize management functions by dragging the column headings

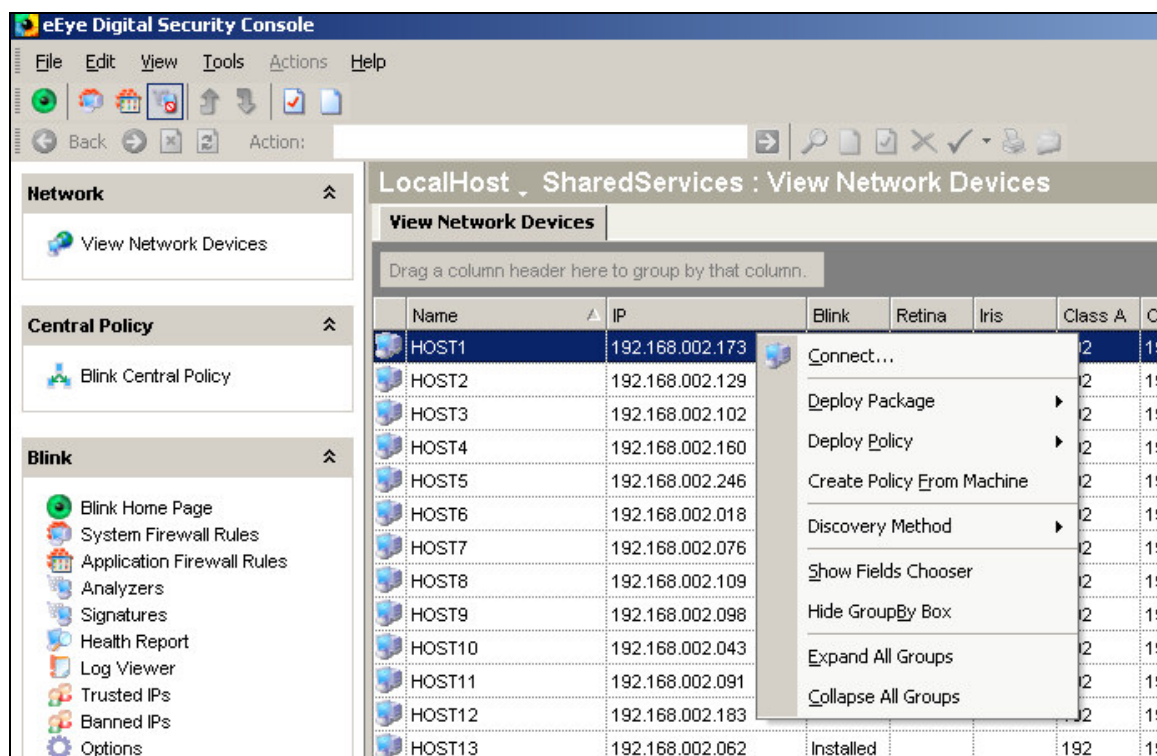
You can also define Blink Groups in the V2.0 and later Discovery View. For example you can define sets of hosts for “Accounting”, “Marketing”, “Chicago”, etc ... and add hosts to each group.

If you want full control over the Discovery Process you can define a set of hosts to import using the “Host Import” feature.



Network Tasks

The pop-up tasks menu that is created by a right click in the Discovery Window is shown in the figure below.



The following table contains the key task options you will want to discuss in a pilot:

Menu Option	Description
<u>C</u>onnect	Select this option and the Security Console connects to Blink installed on the specified workstation. Once you are connected to the workstation, you can make changes to the policy that the application adheres to for the specific installation. This allows you to change settings for the individual policy without affecting the policy of any other installations.
<u>D</u>eploy Package	Select this option to deploy a remote installation package to the selected workstation or group of workstations. To use this command, you must either have an existing package to deploy or you must create a new package to deploy to the selected workstation(s).
<u>D</u>eploy Policy	Select this option to deploy a new or revised Central Policy and replace the existing Central Policy on the selected remote application installation. To use this command, you must either have an existing Central Policy to deploy or you must create a new Central Policy to deploy to the selected workstations.
<u>C</u>reate Policy From Machine	Select this option and the Console will use the Central Policy from the specified machine as the new Central Policy within the Security Console. A dialog box appears prompting you to enter a name for the Central Policy that is saved for future use through the Security Console.
<u>D</u>iscovery Method	Select this option then select one of the following methods that you want the Security Console to use to discover your network's assets: <ul style="list-style-type: none"> ● All Devices – use all discovery techniques ● Active Directory

- ARP
- Netbios

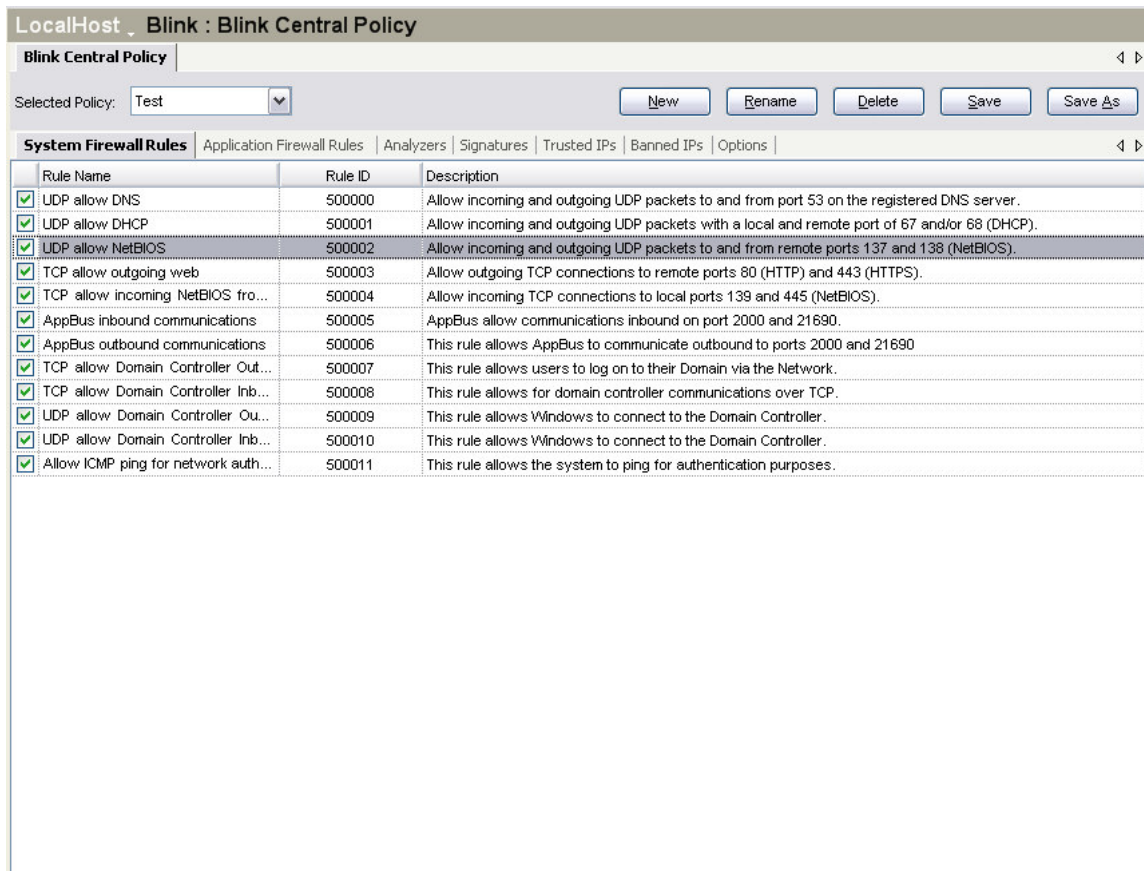
Central Policy View

Overview

The **Central Policy** view allows you to manage (create, edit, delete) policy configurations for Blink agents. The Central Policy view displays a tab for each functional area that a policy is comprised of.

(The tabs are the same categories as the shortcut bar above had.)

A good way to explore or explain Blink policies is to start by creating a policy. Select "Blink Central Policy" and then the "New" button.

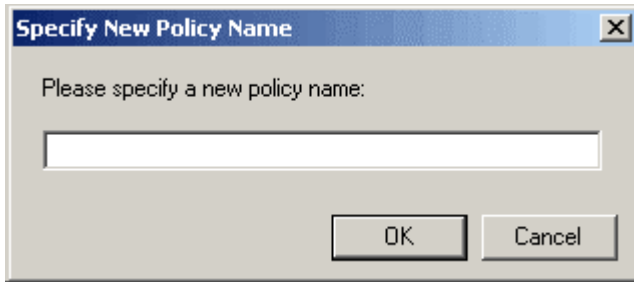


Creating a Central Policy







When you create a Central Policy, the default application settings are automatically, i.e. initially, applied. You can customize these default settings to best fit the requirements of your environment.


Policy Name: Administrators will typically create a custom Central Policy for each specific group (often by department) or by application users, etc.... For example, you can create a custom Central Policy for the sales department, and another Central Policy for customer support, and so on.

1. Click **New** from the **Central Policy** options.
Choose a name that would define a set of hosts in the test environment in the **Specify New Policy Name** dialog box which appears as shown in the following example:



2. Next you can review the settings for each tab and discuss the benefits. The key features and benefits of each tab are outlined below.







Option	Feature/Benefit
 System Firewall Rules	These rules allow/deny port access. They are similar to ZoneAlarm, BlackICE and Sygate. Look at the easy to use wizard for creating new rules. (In the rules: Right Click -> Add Rule)
 Application Firewall Rules	Likewise similar to ZoneAlarm, or Entercept etc... these rules dictate what applications can or should access the network.
 Analyzers	Data (payload) analysis rules versus brute force signature analysis rules are produced by eEye’s Research Team. This is a large part of the value in Blink. Competitive kernel hookers such as CSA catch the problem when the problem is in Kernel memory, which is too late! The damage to the process is done. Blink catches the problem before it hits the kernel where it detects the attack in the packets.
 Signatures	Click this option to display the Blink IPS signatures for the machine that your Security Console is connected to. You can also modify and add to Blink IPS signature tasks from the Security Console such as a signature in email to look for competitive information leaking out of the organization.
 Trusted IPs	Click this option to display the trusted IPs for the machine that your Security Console is connected to. These rules override all other rules. A good example of where this is important would be a patch server where you want to be sure that communication is not interrupted. Set the server to “Allow”. Competitors such as CSA cannot do this and have can block critical patching efforts; thereby adding to security vulnerability problems.
 Banned IPs	Select this option to display the banned IPs for the machine that your Security Console is connected to. You can use this to completely cut off an attacker.

 Options	<p>Under this tab in the "General" category you can "open up" Blink so that rules will not be enabled. This is good when starting in a production environment where you do not want to risk the mistake of breaking applications when the Blink policies are suddenly applied. Later you can slowly enable the rules section by section when the agents are installed. Other items for discussion are the packet capture, logging and event destination settings that you can do under "Options".</p>
-------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Package Creation and Deployment

Once a Policy is defined you need to deploy it with the Blink software to a host that will become a Blink Agent. To perform this, use the "Tools View"

Select **Package Editor** to show the creation of a deployment package containing the policy you just created.

Option	Description
 Common Configuration	Click this option to view and modify the Proxy Server and User Authentication settings
 REM Client Options	Click this option to view and modify the Event Reporting Workgroup and the REM Events Server settings
 Script Editor	Provides an interface to write VB scripts that automate repetitive tasks such as pushing policy updates to other machines.
 Package Editor	Select the Package Editor to view, create, or manage deployment packages.
 Queue View	Click this option to display the deployments currently taking place.
 History View	Click this option to display a history of all deployments.

Open the **Package Editor** and create a new package (agent configuration) for deployment to a set of workstations that you will select from the Discovery View.

While creating the package you will be able to set application parameters, where to poll for central policies, where to report events to, etc...

Key Package parameters are:

- User Mode – what the users at the workstations will experience
 - Interactive – query the user every time an event (Blink wants to block traffic) occurs
 - Alert Only – shows the user a popup warning that traffic has been filtered
 - Silent – the user will not be aware that any traffic is being filtered

- Central Policy
 - Where to gather Policy updates from
 - Policies can come from:
 - A Security Console
 - A Web server via HTTP or HTTPS
 - An FTP server
- Target installation directory on the agent
- Licensing and registration information
 - Licenses are pooled which makes license management simple
- Integration parameters for connecting to REM (optional)

Deploying a Blink Package to Target Hosts

Overview

- To deploy a package, select a set of hosts in the **Discovery View** and **Right Click** to choose Deploy Package
 - Note: The initial package deployment requires that file sharing is enabled. Thereafter Blink uses its own secure SSL communications channel, but the initial deployment from the Security Console requires IPC\$ sharing. See Troubleshooting below for details.
 - The **Security Console** allows you to deploy applications from a central location to other workstations that there are licenses for.
- When you deploy a package, the application with its policy is installed on the selected workstations that you chose from the Network View. (You must create at least one policy before deploying your first package.)
- Updates to the Agent's Policy are applied when the agent checks the Security Console location for new updates to the policy that they are subscribed for.
- Use the **Queue View** to see what packages are queued to be deployed
- Use the **History View** to see what packages have been deployed
 - Packages will be removed from the Queue when they are deployed and will be recorded in the History View
- You must have Domain Administrator rights to the remote workstation(s) that you want to deploy the package to. When the deployment starts you will be prompted for credentials if the user does not already have remote administrative privileges .



Demonstrate the Package/Agent that has been Deployed

1. Connect to the agent through the Discovery View by Right Clicking the host
2. Review the Policy by selecting the Blink Home Page
3. Review the Retina Vulnerability Scanner

Note that scalability is achieved by compartmentalizing Blink Security Console installations. We recommend that you manage approximately 1000 hosts from each Security Console if you desire to maintain the ability to rapidly swap policies on all managed hosts. Otherwise you can manage as many hosts as you desire from each Security Console. In this case the limiting factor will be the ability to organize the managed hosts in the Discovery View's tree component.

Demonstrate Deployment of a New Policy

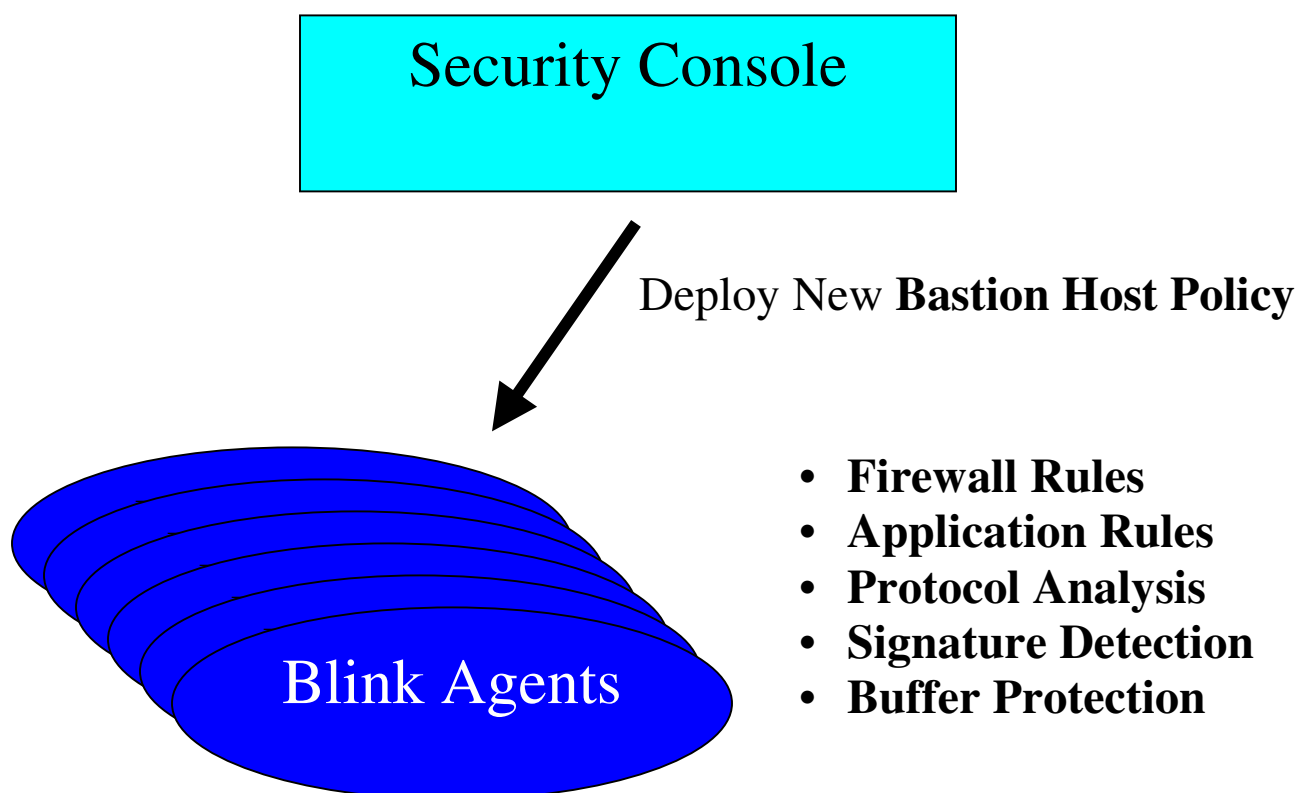
You can update the policy (the set of rules that control Blink of an Agent) immediately.

We have already deployed a Package that contained a Policy. Now we will update the Policy. (not the software package, but the Blink rules set, called a Policy).

1. To immediately deploy a new Policy select a set of hosts in the Discovery View
2. Right click and select Deploy Policy
3. Select the new policy

The following is a scenario to consider. Since the Security Console can be used to rapidly change policies, use it to temporarily lock the network down in times of high risk.

Scenario: The network risk suddenly increased. Deploy a "Bastion Host Policy" to all hosts!



After deploying a "Bastion Host Policy" which locks the host down the eEye Blink agents now monitor traffic and strictly secure the network until the danger level decreases. This mass policy change or swap is possible because the console has the ability to update the agents on demand.

Otherwise the agents are continually polling for new changes to their policy as adjusted by the Security Administrators.

Conclusion

Hopefully the pilot has taken you through the fundamental benefits of Blink. The layers of security protection that you should have an understanding of are:

- **An intrusion prevention layer that shields the asset from unknown attacks without the use of signatures**
- **A rules-based intrusion prevention layer to pinpoint and shield from known attacks**
- **A network-level firewall to control the unauthorized connectivity of the asset from others**
- **An application-level firewall to prevent unauthorized programs from running on the asset**
- **A Retina based host-level vulnerability assessment scanner to detect and report known security issues on the asset.**

These 5 features of Blink make it a superset of other products on the market. Blink has depth and breadth which make it an excellent choice for Host Based Security.

As noted earlier Blink scalability is achieved by a “Divide and Conquer” approach to Blink Security Console installations. The number of installations can be infinite all reporting to REM consoles. In the case that you want to be able to rapidly change host policies in emergency conditions, we recommend that you manage approximately 1000 hosts from each Security Console. Technically you can manage as many hosts as you desire from each Security Console, but you may find that the limiting factor is the ability to organize your hosts in the Discovery View’s tree component.

These 5 layers of defense, managed from the central Security Console, with unlimited scalability, enable users to protect an enterprise of hosts.



Blink Installation Process Template

Overview

Credits: This template section was originally written by Celinda Garza for the Station Casinos pilot.

It is intended that Sales leverage this section. If this document is in Adobe, you should ask for the Word version so that you can cut-and-paste the material into your documents.

Phase One - Planning

The initial phase of the pilot engagement is to determine the unique challenges your organization faces, to gain a better understanding of the business drivers leading you to consider our solution. An outline of the network topology, the operating systems and applications being used should be assembled.



Phase Two - Build Objectives & Agreement

Determine the agreed upon objectives and success criteria for the Proof of Concept.

Phase Three - Proof of Concept, Technology Evaluation

eEye Digital Security will assist with the deployment and initial management of pilot engagement as the customer applies the Blink technology to their environment.

Phase Four - Review of Findings

Upon conclusion of the pilot engagement and consultation with customer, the customer will deliver a document to eEye Sales stating the effectiveness of the pilot and the degree to which the success criteria were met.



Pilot Success Criteria Strategy

The success of the pilot program should be judged in relation to the objective and measurable criteria outlined below. The following items were chosen as they have been determined to encompass critical aspects of the solution's functionality and will provide an objective measurement of the product's capabilities.

The primary measure of the success of the pilot program is the deployment and testing of Blink on an appropriate and relevant number of devices within the client's network. Therefore, eEye requests that the client commit to the availability of such devices in the case of an on-site pilot supported by eEye and/or its partner representatives.

Scope of Deployment

- The client agrees to allocate the following number and type of devices to the first phase of on-site deployment
 1. _____ workstations running Windows version _____
 2. _____ workstations running Windows version _____
 3. _____ Laptops running Windows version _____
 4. _____ Servers running Windows version _____
 5. _____ Servers running Windows version _____

Installation

- Installation of administrative tools
- Review deployment plan according to customer input
- Define security policy to be deployed via console
- Implement security policy in Blink Console

Deployment

- Deploy Blink with custom policy to single remote test machine



- Verification of deployment of policy to remote machine using Blink Console
- Manual, visual verification of deployment on test machine

Management

- Remotely deploy Blink agents with aforementioned security policy
- Perform remote un-installation of single Blink agent using Blink Console
- Perform policy change, deploy policy to all devices and verify change via Console
- Initiate simulated attack(s) on at least five (5) machines and verify attacks have been blocked and logged on each individual machine by accessing machines via console



Additional Client Specified Criteria

- Criterion:

- Criterion:

- Criterion:

Blink Q+A and Tips

1. I want to update the Blink product asynchronously from a command line.

- You can run C:\Program Files\Common Files\eEye Digital Security\SyncIt\SyncItGUI from the shell
- Review debug_syncIt.log for detailed status of the update.

2. How do I get a quick hands-on review of the Blink features.

- Run Blink in standalone mode on the local host by installing the product locally.
- Review the features at the top level, especially:
 - **Intrusion prevention using:**
 - **signatures**
 - **and protocol analysis**
 - **Network-level firewall to control IP address and port access**
 - **An application-level firewall**
 - **Retina based host-level vulnerability assessment**



3. I want to know how the hosts are being discovered :

To learn how Blink discovered a host, look in C:\Program Files\Common Files\eEye Digital Security\Shared Services Host\data\Discovery.xml . At the top you will see the "id" for each of the discovery methods.

For example:

```
<method id="0e207c4f9aad4df5882e4b5fc8437e82">  
  <name>Active Directory Method</name>  
  <description>Discover computers using Active Directory</description>  
</method>
```

For each host you will see a list of "ids" in the "<methodItems>" tags

4. How do I remotely debug Blink?

- As admin you can use the console to connect to the debugged machine and see the logs, policy settings and product configuration.
- Or you can switch Blink into a non-Silent mode temporarily and have the user run the Blink gui from the start Menu and work through issues over the phone.

5. How do I deploy just the Blink standalone application?

- Run the Blink setup program on the host.
- You will need a license from eEye
- The installation only takes a few minutes.

6. I don't want updates coming directly from eEye. How do I distribute them from my server?

- Show Options -> Common Configuration which allows you to set up a proxy server from which you can update the software.

7. How do I check if all packages arrived at the target host, and their history of deployment?

- The History and Queue Views in the Security Console show what it executed
- Each target host has the directory C:\Program Files\Common Files\eEye Digital Security\Shared Services Host\data\Packages that contains a history of each package and the detailed settings of that deployment.

8. How do I enable or check IPC\$ file sharing from my server?

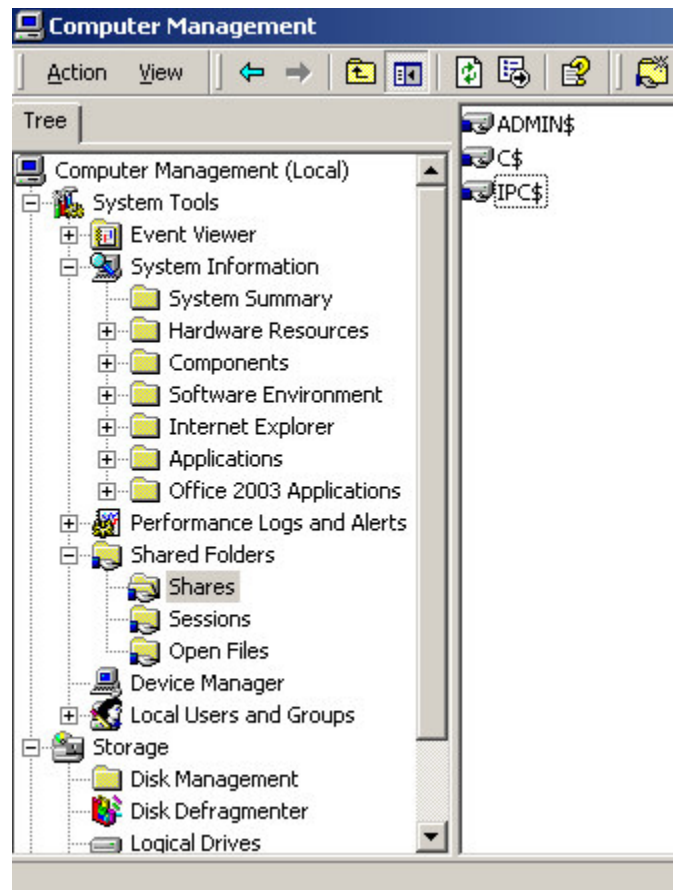
1. Double-click **My Computer**.
2. Double-click **Control Panel**.
3. Double-click **Network and Dial-Up Connections**.
4. Right-click the network connection you want to change and press **Properties**.



5. Check the **File and Printer Sharing for Microsoft Networks** box on the **General** tab.
6. Press **OK**.

Or you can perform:

- Under Computer Mgt -> File Sharing look for IPC\$. You can Right Click to enable or disable file sharing.
- If File Sharing is against your security policy you have the following options for deployment:
 - Enable Sharing temporarily for the initial deployment
 - Use your third party tool that has software deployment agents installed on each host.
 - Install Blink manually on each host; thereby enabling central deployment of software and policy updates thereafter.
 - Writing a login script so that users can receive the Blink package upon login.



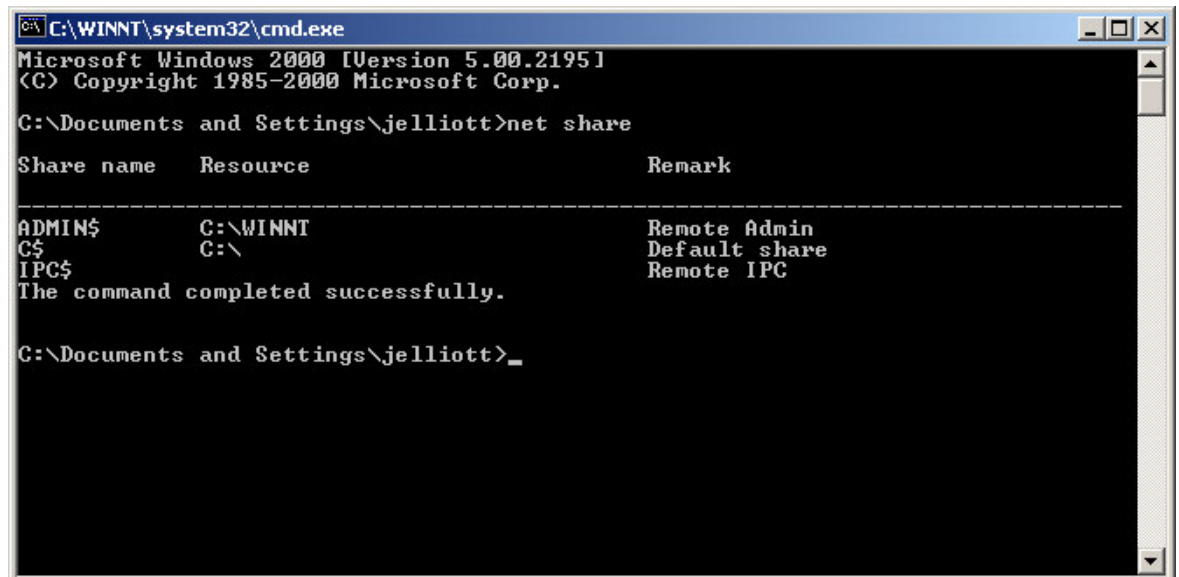
Under Computer Mgt -> File Sharing you will see IPC\$ sharing.



Another way to change File Sharing (in XP Pro):

1. Click *Start | My Computer | Tools | Folder Options | View*.
2. Scroll to the bottom of the list of advanced settings and check *Use Simple File Sharing*
3. Click *OK*.

Note that the command line can also determine File Sharing by typing "net share". Look for IPC\$. The "share" and "view" commands can



Or programmatically you can do the following:

You can also use the following scripts to temporarily share a host so that you can deploy the Blink agent. Note that once the agent is deployed, file sharing is no longer required.

```
Const FILE_SHARE = 0
Const MAXIMUM_CONNECTIONS = 2
strComputer = "targetBlinkhost"
Set objWMIService = GetObject("winmgmts:" _
    & "{impersonationLevel=impersonate}!\\" _
```



```
& strComputer & "\root\cimv2")
Set objNewShare = objWMIService.Get("Win32_Share")
errReturn = objNewShare.Create _
("C:\", "MyShare", FILE_SHARE, _
MAXIMUM_CONNECTIONS, "Public share for the Finance group.")
```

```
Const FILE_SHARE = 0
Const MAXIMUM_CONNECTIONS = 2
strComputer = "targetBlinkhost"
Set objWMIService = GetObject("winmgmts:" _
& "{impersonationLevel=impersonate}!\\" _
& strComputer & "\root\cimv2")
Set objNewShare = objWMIService.Get("Win32_Share")
errReturn = objNewShare.Create _
("C:\", "MyShare", FILE_SHARE, _
MAXIMUM_CONNECTIONS, "Public share for the Finance group.")
```

9. How can I add some of our in-house scripts to Blink?

- Goto the Options -> Script Editor to see how scripting can be integrated into Blink. Competitors do not provide this flexibility.

10. Can Blink be used for quarantining?

- Yes you can select a set of hosts and immediately deploy a set of harsh policies that will lock them down. Our roadmap has the enhancement of making this fully automated and based on any number of inputs.

11. What will Blink do in our environment?

- You can set Blink to run "Passive Mode" which means that Blink will not suppress any traffic but will log all events. This way you can get a preview of what sorts of actions Blink will take without having Blink actually take them.
- To set "Passive Mode" set 2 checkboxes. Go to Options->Advanced->
 - Enable Firewall Passive Mode
 - Enable IPS Passive Mode

12. How do I configure Blink for Exchange Server?

You can set Blink to run "Passive Mode" which means that Blink will not suppress any traffic but will log all actions it will take when enabled. You can use this feature to test your Exchange configuration.

Exchange 5.0 supports POP3 to retrieve messages from a mail server. Other mail clients in your enterprise may be Internet Mail and News, Windows CE Inbox, and Internet Mail Service for Windows, with clients such as Pegasus and Eudora Pro.



POP3 clients use TCP port 110. Exchange Servers listen on this port for incoming connection requests from the POP3 clients. SSL (Secure Sockets Layer) authentication uses port 995. Therefore, you should configure the Blink firewall filtering to include TCP port 110 or TCP port 995 for POP3.

POP3 to SMTP (Simple Mail Transfer Protocol) communication is over TCP port 25.

Exchange version 5.5 supports IMAP4, the Internet Message Access Protocol, which is a superset of POP3. When using Basic or NTLM authentication and TCP, the IMAP4 server listens on TCP port 143. If SSL authentication is used, TCP port 993 is used. Router and firewall setups should therefore take into consideration the access to TCP port 143 or TCP port 993 when this protocol is a supported feature for messaging.

For an LDAP client to connect to an Exchange Server the ports that need to be configured on the Blink firewall are for the authentication. With Basic authentication, that is port 389. For SSL, the Exchange Server computer listens on is 636.

These are the most common protocols used with Exchange. Note that there are other applications that can connect to your Exchange Server if so configured. If you are not sure what these are, set Blink to run "Passive Mode" to see what traffic is being inadvertently blocked. Then enable Blink into "active mode" so that it can start performing its job.

13. How do I configure Blink for a Domain Controller?

After doing an initial firewall setting, set Blink to run "Passive Mode" to see what traffic is being blocked.

Use the following setting initially.

For NT set Blink to allow ports:

- 42/TCP - WINS
- 135/TCP - RPC
- 137/UDP - NetBIOS Name
- 138/UDP - Netlogon
- 139/TCP - NetBIOS Session

For Win 2K and up:

- 53/both - DNS
- 88/both - Kerberos
- 135/TCP - RPC
- 389/both - LDAP
- 445/TCP - SMB
- 636/TCP - LDAP SSL
- 3268/TCP- LDAP GC
- 3269/TCP- LDAP GC SSL



14. How do I know what Blink is blocking?

- You can set Blink to "Log Denied Traffic" which means that Blink will write into its event log any traffic that it stops. This will give you a record of what actions Blink is taking behind the scenes.
- To set "Log Denied Traffic" set the checkbox. Go to Options->Advanced->
 - Log Denied Traffic
- **Note that you should only use this setting for testing.** This generates a large number of log entries. Especially do not run a scanner, such as Retina, against the host and expect to be able to review the thousands of entries in the log.

15. How can Blink capture all events defined in rules?

You can set a registry key to enable the capturing of all triggered events.

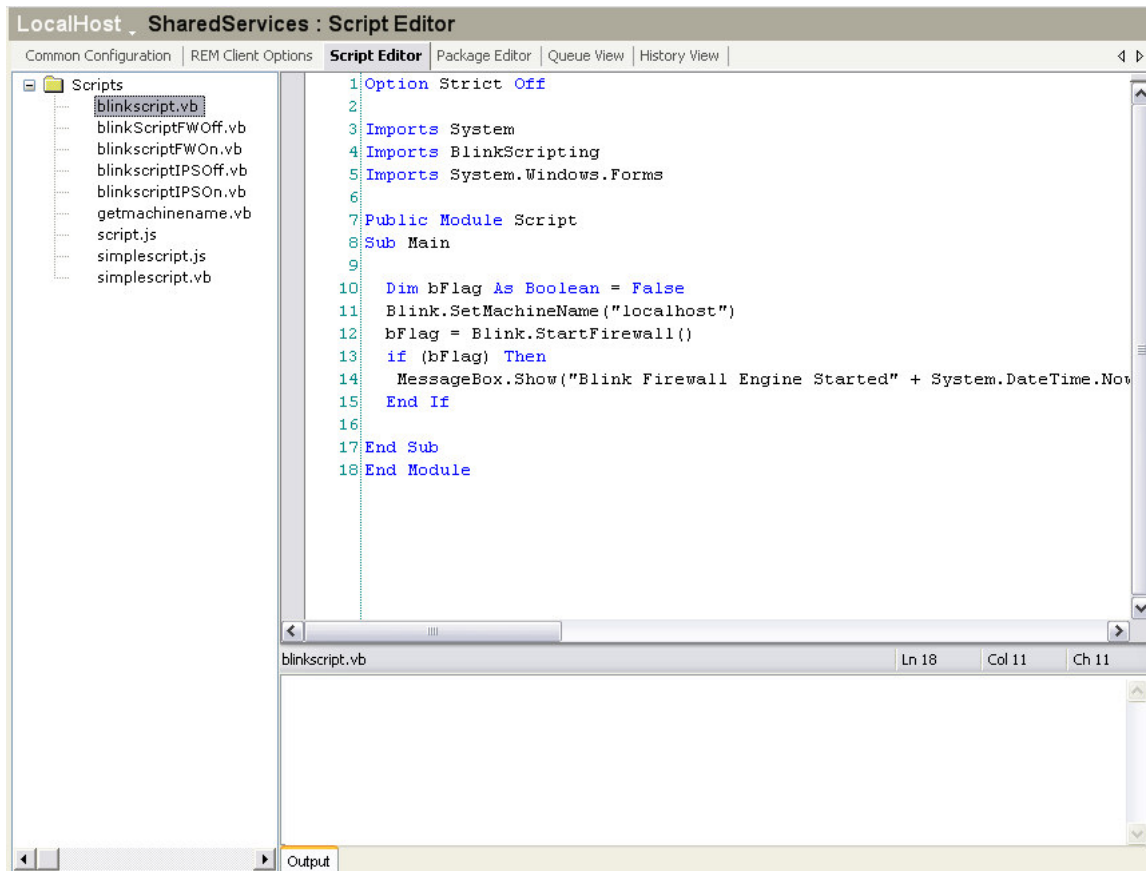
```
[HKEY_LOCAL_MACHINE\SOFTWARE\eEye\Blink]  
"ForceAllEventsLogging"=dword:00000001
```

16. How does Blink stop Shareware?

- Lock down your host applications by setting a policy that only allows the applications you have installed to access the network. New shareware applications, or ones already on the host will then be denied access to the network.
- You can also restrict ports and if possible IP address access by configuring the Blink Firewall Rules if you know the parameters for the shareware you are trying to stop..

17. How can I include scripts in Blink?

Console scripting enables a user with the correct credentials to script tasks for a client machine locally or remotely. In this manner scripting can be run at the command prompt, from the desktop or as a scheduled task. The scripts can be written in either VB .NET or javascript. If they are written in VB .NET the script editor within the console allows you to compile the scripts and run them.



18. Can Blink secure my wireless laptops?

Yes, Blink will run on wireless workstations. You can create policies that allow you to control security settings based on the Access Points that you are using.

19. How does Blink help me comply with Sarbox 404?

Blink is arguably the single most comprehensive Section 404 tool for digital security that an organization can utilize. Under Section 404 of Sarbanes Oxley, the management team of a public company must assess and control the security of the company's financial reporting. Specifically Section 404 says that the management commission for Sarbox compliance must:

*"1) state the responsibility of management for establishing and maintaining an adequate internal control structure and procedures for financial reporting; and
(2) contain an assessment, as of the end of the most recent fiscal year of the issuer, of the effectiveness of the internal control structure and procedures of the issuer for financial reporting.*

"

The internal controls assessment must show the ability to:

1. Detect problems with data alteration and general security.
2. Analyze overall data security.
3. Protect and maintain data validity.
4. Remediate data integrity issues within a repeatable process framework.

Translated, this means that the company needs to secure hosts by protecting, analyzing and remediating their vulnerabilities in a repeatable, controlled manner.

Blink provides the features and a framework for doing these tasks. Blink detects and protects against real-time attacks. The Blink vulnerability assessor, that uses eEye's Retina technology, enables organizations to analyze and remediate security issues. Therefore, from a digital security standpoint, Blink provides the functions set forth by Sarbox 404.

eEye advises that an enterprise deploy Blink to meet the Section 404 functional requirements for internal controls and periodically produce reports using the eEye REM console to establish and prove that due diligence is being performed. Blink and REM together provide the required functions and controls necessary to automatically and repeatedly process the requirements dictated by 404.

20.Does Blink run with NetWare?

Blink will run on NetWare workstations (clients), but not on NetWare servers.

Here are the TCP and UDP ports used by NetWare 5 for Pure IP connectivity:

- TCP 524 - NCP Requests - Source port will be a high port (1024-65535)
- UDP 524 - NCP for time synchronization - Source port will be a high port
- UDP 123 - NTP for time synchronization - Source port will be the same
- UDP 427 - SLP Requests - Source port will be the same (427)
- TCP 427 - SLP Requests - Source port will be the same (427)
- TCP 2302 - CMD - Source port will be a high port
- UDP 2645 - CMD - Source port will be the same (2645)

Best thing to do is start with everything open and then apply the Application firewall rules.

IPX traffic will not be affected nor processed by Blink since IPX is a legacy protocol and not based on TCP/IP.

21.Does Blink run with Windows Terminal Server and Citrix?

eEye does not allow multiple copies of blink/console to be opened on the same machine.... This is something short-term though and in the long term we will plan to support multiple copies/users of blink on a single machine.



It should be noted though that Blink is still a valuable solution for Citrix and terminal servers. The IPS, System Firewall, and Vulnerability Assessment, will still properly function and protect the machine. The application firewall also does work, however depending on who opened blink first, they will be the one that gets the prompts; however we typically suggest turning prompting off anyways.

22. What is Blink Application Protection?

Blink provides application protection from attackers trying to exploit buffer overflow vulnerabilities. It will detect remote payloads attempting to run in memory not intended for code execution such as the stack, heap, or in a DLL data section. If a malicious input overwrites the portion of the stack and the return address with a clever piece of machine code, Application Protection signals the kernel to stop and the program is terminated. Application Protection will cause the process to terminate instead of executing the payload code; thereby averting a compromise.

See Appendix A for Application Protection configuration.

23. What is Blink API Protection?

Blink provides API protection from code that tries to intercept Win32 API calls. Injecting a DLL into the address space of an external process is a primary technique for spying on a host. It provides the ability to inspect the actions of all of a process's thread activities. Blink provides injection protection by watching for hooking activity. It will detect common hooking techniques used by attackers attempting to insert themselves into code using calls such as "SetWindowsHookEx".

For example to use SetWindowsHookEx and hook the user's keyboard, attack code could do this:

```
hook = SetWindowsHookEx( WH_KEYBOARD,
                        myhookprocedure,
                        hinstance,
                        NULL);
```

The "myhookprocedure" is the malicious "shim" or "proxy" code that then processes the keyboard activity and sends the results to a foe=reign host, etc.. The callback would look something like this:

```
KEYDLL3_API LRESULT CALLBACK myhookprocedure(int ncode, WPARAM
wparam, LPARAM lparam)
{
    ProcessTheKeyStroke(hwnd, WM_USER+755, wparam, lparam);
    return ( CallNextHookEx(hook, ncode, wparam, lparam) ); //pass control to normal,
expected procedure
}
```

The last line calls the code the application expects. Thus the malware is inserted into the keyboard activity.



lparam, which is passed into the hooking procedure is a structure that contains the "scanCode" element which indicates what key was pressed.

The Import Address Table (IAT) contains the addresses of all functions, imported by a module. A common hooking technique is to locate the IAT of the target executable module and modify its entries so that calls to imported functions are redirected to an attacker's code. The attacker then makes it look as if nothing malicious has occurred by passing control on to the original function. So everything behaves normally to the end user of the application. However in reality every call to an imported API will call malicious code first.

To overwrite IAT entries, the attacker obtains the name and address of IAT entry for the given imported function:

```
WriteProcessMemory(GetCurrentProcess(),IATentryaddress,&ptr,4,&newaddress);
```

WriteProcessMemory can also be used to directly alter a process's binaries. When API Protection is checked, Blink watches for these calls as indications of malicious activity.

24. How does Blink work with EAP, EAPOL, 802.1X and RADIUS ?

First a quick overview of these protocols to understand Blink's role:

802.1X is an attempt to standardize the encapsulation of authentication protocol. All IEEE 802 media, such as Ethernet, Token Ring, FDDI, RADIUS and 802.11 wireless LANs may use IEEE 802.1X to enable authenticated access. 802.1X encapsulates the Extensible Authentication Protocol (EAP) which is used for passing authentication messages. 802.1X typically connects a client to wireless access point. **802.1X does not perform the actual authentication.** When utilizing 802.1X, you need to choose an EAP type, such as Transport Layer Security (EAP-TLS) or EAP Tunneled Transport Layer Security (EAP-TTLS), which defines how the authentication takes place. There are many EAP types that may be encapsulated in 802.1X, which itself may be encapsulated in Physical Layer protocols such as Ethernet, Token Ring, FDDI, 802.11, etc...

802.1X authentication for wireless LANs has three main components:

- The **supplicant** (usually the client software);
- The **authenticator** (usually the access point);
- The **authentication server** (usually a Remote Authentication Dial-In User Service server, although, RADIUS is not specifically required by 802.1X)

EAP messages pass between the supplicant and authenticator and from the supplicant to the authentication server (via the authenticator). EAP messages from the authenticator to the authentication server typically use the RADIUS protocol.

Blink coexists with 802.1X as it does with many protocols such as SNMP, FTP, RPC, etc... Possibly confusing, is that there is a specific Juniper VPN integration to Blink that we offer.



This integration influences Juniper's 802.1X/EAP like VPN access control protocol by providing the VPN switch with information about the host's status. The switch then uses this information along with the Juniper authentication exchange to decide whether to allow the host to enter the network.

With the state of the technology today, each vendor that uses 802.1X, or their own protocol, will have effectively proprietary authentication to which there will be API interfaces. eEye plans to interface to the industry leaders as we have to Juniper. Additionally we are adding interfaces to Blink that will enable custom integrations to be performed.

25.How does Blink interoperate with TCB ?

It is with relief that the TCG was formed to try to take the proprietary nature out of the security arena so that security products can interoperate and complement each other. For example the TCG's Trusted Network Connect (TNC) working group is trying to standardize the issues surrounding when and how a host can access a network. Today there are multiple technologies such as NAP and NAC among others, that address pieces of the problem, but are proprietary duplicate interfaces, each requiring a separate interfacing effort. The TNC is focused on standardizing access protocols so that disparate security products can assess and protect hosts before, during and after connecting to an enterprise switch.

The TCB TPM baselines platforms by taking an SHA-1 hash of all executable code or data . Data deviations result in a different hash value, so that malware affected applications or unauthorized data can be detected. Blink complements this by detecting new malware and exploit data that TPM has not yet been able to baseline. For example when a user downloads a file or views attacking Javascript, Blink stops the attack. The desired file additions, filtered by Blink, can then be added to the TPM baselines. The bad files are stopped.

The TCG is also defining standards for how hosts with authenticate themselves. This will work well in controlled environments and will aide in reporting on communication activity. As this identity management grows over the next decade it will provide a better forensic path for determining who talked to whom and pinpoint the origins of malware. Until this infrastructure is completely established we will continue to attach to unknown hosts such as the many web servers we visit every day. In the mean time Blink will be protecting hosts from exploitation. Even after all of today's existing hardware has been disposed of, and TPM hardware is prevalent we will still be challenged by the decision to accept or not accept a "certified" connection. It is analogous to having license plates on cars. It is a good form of identification, but that does not completely thwart illegal driving. There will be "certified TCB hosts" spreading malware. Blink will provide the real-time protection needed.

26.How can I use Blink stop Spyware and Malware?

- Blink V2.0 has a database of known spyware signatures which it uses to scan memory running active processes and the system's hard disk. Upon detection the spyware is either quarantined on the disk or if active in memory, removed from memory by killing the process or the thread that is running the detected spyware.



You can use the directory dialog to choose the malware quarantine location. You may also want to periodically change the locations so that you can organize the malware into different directories based on time periods.

- Spyware must talk back to the “mother ship” host it is logging to. Blink can detect new spyware ports opening to talk back to the “mother ship”.
- If the spyware “tunnels” through another protocol, Blink signatures may be created to detect the spying payload content leaving the host.
- You can also restrict known spyware ports and if possible IP address by configuring the Blink Firewall Rules.
- When spyware is detected Blink can capture the full session so that forensics may be performed. As we know often the attackers are elusive and use temporary sites, which at least can be shut down, but in most cases spyware traffic is returned to established sites and businesses that can be warned with Blinks’ forensic proof to backup the warnings.
- Spyware installation is blocked by Blink’s Application Protection (code named Kevlar) which protects Blink from all buffer overflow attacks. If the user explicitly installs spyware, the above detection and removal mechanisms will stop the exploitation of the host.

27.How do I allow applications to run that look like Spyware or Malware?

Blink has a kernel mode API hooking driver that is used primarily to detect and stop process code injection attempts.

However some legitimate applications need to use similar techniques to operate. To exempt these applications from Blink’s API protection engine, API rules may be defined by the user.

API rules are defined in the text file named Apiext.ini; found in Blink’s home folder, The format for creating a rule are as follows:

#process;MD5;Protection method;action

Process

Can be a process name, a full path to the process, a full path containing environment variables, or empty.

MD5

If present, the process field will be ignored.

Protection Method

Can be one of SetWindowHookEx, TerminateProcess, WriteProcessMemory

Action

Set to 0 means Disable check



Set to 1 means Enable check

Example Rules:

```
#disable SetWindowHookEx for all proceses  
*;;SetWindowsHookEx;0
```

```
#stop the user from forcibly terminating a process  
%ProgramFiles%\company\program.exe;;TerminateProcess;1
```

28.How does Blink stop Phishing and Identity Theft?

Phishing is detected by the POP3, IMAP and HTTP protocol analyzers. When a link is detected that has text and the text of the link appears to be a URL that is different from the URL actually linked, then an alert is triggered (#7001) and the link is replaced with the non-hyperlink text "[LINK: <address>]" where <address> is the actual destination of the original link. This disables the user from following the phishing link to the fraudulent server. The phishing alert in Blink's logs indicates both the text and the actual address that the text linked to.

29.How much memory does Blink consume?

On an XP SP 2 with 386 MB of RAM Blink takes a mere 15 MB of physical memory. The physical memory plus virtual memory on XP 2 sums to 65 MB.

30.How much CPU does Blink consume?

With no network traffic Blink uses 0% CPU. When loading typical web pages Blink uses 2-3% of a P4 1.8GHz with 1GB RAM running Windows 2000 SP4.

31.How do I use a proxy server for Blink licensing?

Normally the Security Console communicates with eEye's server to coordinate licensing. If the Console administrator prefers to use a proxy server to make this connection to eEye they may do so in the Security Console's "Common Configuration" area. There is a text area for defining the address, username and password for the proxy server.

32.How does Blink stay up to date?

Blink has a "Sync-It" utility imbedded in the distribution that automatically updates the Blink agent and console programs. Within these updates are new Blink Rules, based on our research team's findings. This way customers stay up to date with the latest protection without any actions required. The "Sync-It" utility is scheduled by the user, for example mid-night every other day, and the updating process will take place in the background.

33.Where are we today, and what is the roadmap?

- Today Blink is the most comprehensive host security solution on the market. Blink does proactive vulnerability assessments, real-time traffic analysis and blocking, and has the ability to perform forensics through its packet capture facilities. eEye will continue to add to this feature set so that Blink remains the best of bread in host security.



- From a scalability standpoint Blink is managed from the Blink Security Console. This approach works well for large networks due to the completely flexible nature of the Console. There is not limit to the number of hosts that the Security Console can manage. As many Blink agents as are practical to manage from one location is the only limitation.
- In a nutshell, Blink is ahead in features and has the ability to scale; however there is more to come. Future efforts will concentrate on:
 - Blink reporting – more event correlation and more charting.
 - More anti spyware, malware and phishing modules
 - Anti-virus protection
 - Quarantining and policy swapping based on the environment
 - More integration with Active Directory
 - Integration with more third party platforms so that existing investments can be leveraged

34.How should I update my agents after they have been deployed?

There are several approaches that you can take:

- 1) Let the SynchIt update process which comes from eEye do the update.
- 2) Update the Security Console, run uninstall package from the SC for all targets, create a new package, and deploy the package.
- 3) Update the Security Console, then run ForceSynchItUpdate.vbs from the SC. You will have to write a loop in the vbs for the host addresses to deploy to.
- 4) Install a REM Update server. Get the new Blink update from eEye. Have the agents update from the REM Updater.
- 5) Use the Third Party Deployment Tool.

35.What processes does Blink run?

eeyessh.exe	- Shared Services: does deployment, discovery and Policy
management	
eeyeevnt.exe	- Runs the Application Bus communications for eEye products
eeyeab.exe	- Address Book process which performs the ARP discovery
blink.exe	- Main Blink exe for the GUI
blinksvc.exe	- Blink engine that processes the rules.
blinkrm.exe	- Interfaces with configuration data such as the rules data. blinksvc and blink.exe process the rules supplied by blinkrm. Master.xml - has the default rules Group.xml - reflects the Central Policy changes. This also
overlays Master.xml	



Machine.xml - has the user defined rules and rule changes.
This overlays Master.xml and Group.xml

36. Do I need .NET?

The Security Console requires .NET and will automatically install it on the target host. Note that Blink agents do not require .NET.

37. Blink filters on FTP appear to not work

Your intention is to stop any client from issuing the ls command for example, or downloading any file with exe in its name. If you create a new Blink IPS signature, you then choose the FTP protocol, and you choose to scan for a signature within a certain sub-protocol, such as file name, or commands, the signature will only work on an FTP server. The work-around is to create a new signature, based on TCP, choose 21 as the destination port, then add the scanning for the items you wish to have.

38. Troubleshooting: How do I troubleshoot the Blink installation?

- Run installer with the following arguments: *Blinksetup.exe /L*v C:\blinkinst.log*
- C:\blinkinstl.log file will be generated.
- Review the contents or send them to eEye.

39. Troubleshooting: How do I troubleshoot the Console installation?

- Run installer with the following arguments: *Consolesetup.exe /L*v C:\consoleinst.log*
- C:\consoleinstl.log file will be generated.
- Review the contents or send them to eEye.

40. Troubleshooting: Blink Security Console will not start:

If there is a dead instance of a Blink service that is blocking the restart of the service:

- In Task Manager, kill the process "shell.exe"
- Also try killing the Blink services:
 - eEye Blink engine (for Blink Agent)



-
- If all else fails or you do not have Admin privileges reboot the system
-

41.Troubleshooting: Blink application errors:

If errors are reproduceable run Dr. Watson to help get additional information about the error.

When Dr. Watson encounters an error it will be logged under the [file](#) "drwtsn32.log" or "user.dmp" when running [Microsoft Windows NT](#) or [Windows 2000](#). When running Microsoft [Windows 95](#), 98 or [ME](#) the file is logged with a .WLG file [extension](#) and stored under the \Windows\Drwatson or \Documents and Settings\All Users\Documents\DrWatson [folder](#). You can review this file or send it to eEye for analysis.

42.Troubleshooting: Blink agent debugging:

Add DWORD keys named blink, blinksvc and blinkrm under the following branch:

HKEY_LOCAL_MACHINE\SOFTWARE\eEye\Blink

Set the "Value" to:

0 – Debugging disabled

1 – DebugView output

2- A file will be created in Blink's folder having the .log extension

(blink.log, etc)

In the case of a BSOD Crash

In case of a crash, 2 files will be generated in Blink's installation folder: (**crash.xml** and **crash.dmp**).

To View Shadow Rule Creation

Blinksvc.log shows the shadow rules. The reason we didn't add a GUI for them is because of their dynamic nature. Some shadow rules expire in 3 seconds some in 90 seconds and some when the process that created them is getting closed.

Other problems

Install a debug version and use DebugView to generate and save a log file.



43. Troubleshooting: Blink protocol processing debugging:

Failure to start

Blinsvc service will write an error event in **Windows Event Log** in the **Application** page.

To export these entries, right click on the Application label and select Save Log File As

44. Troubleshooting: Blink deployment debugging:

Install package:

A folder named RDLogs will be created in Installation folder (for blink in c:\program files\eEye Digital Security\Blink) that contains 4 log file (debug_syncIt.log, application bus installer log, blink installer log and eeyeremoteinstall service log). If this directory doesn't exist then search for these logs in %Temp% \eEye Digital Security\RDLogs.

Uninstall package:

A folder named RDUinstallLogs will be created in %Temp% \eEye Digital Security folder that contains 4 log file (debug_syncIt.log, application bus installer log, blink installer log and eeyeremoteinstall service log).

45. Troubleshooting: Blink product update debugging:

The updater creates a log file called debug_syncIt.log in C:\Program Files\Common Files\eEye Digital Security\SyncIt.

In case of an interactive run of the updater, the update will generate a xml and a dmp file in case of a crush.

46. Troubleshooting: Blink communications debugging:

For communications issues involving application remote management and Blink central policy:

- Stop Application Bus via Service control panel applet
- Stop anything that uses Application Bus: Retina, SecureIIS, Blink
- Run the DebugRedist.exe setup at [\\builds\store\DebugRedist](http://builds.store.DebugRedist)
- Install the debug version of Application Bus
- Download the free DebugView utility from <http://www.sysinternals.com>
- Launch DebugView.exe
- Start Application Bus
- Start anything that uses Application Bus: Retina, SecureIIS, Blink
- Save log to a file and send to eEye Development

47. Troubleshooting: Blink Security Console debugging:

General Debugging

Create a file called 'Debug.ini' in C:\Program Files\eEye Digital Security\Console

This file should contain the following contents:



[Debug]
Log=1
Return Debug.log file from the same directory

Check HKEY_LOCAL_MACHINE\SOFTWARE\eEye\Application
Bus\Applications\Blink
If the Security Console is missing the Blink management components
in the UI.

If you have "exceptions" please send screen shots of the stack dump
to eEye

Additionally send information in both tabs of the error view dialog

48.Troubleshooting: Blink does not seem to filter my traffic:

In the case that you start Blink on a host that already has established TCP based applications running, you will not see Blink filtering the traffic immediately. Blink applies its firewall rules for TCP traffic only after it sees the initial TCP handshake. (For all other protocols the rules are applied for each incoming or outgoing packet.) Therefore if a connection was established before the Blink firewall was engaged, that connection will continue to function since Blink does not have a chance to catch the TCP handshake and subsequently filter the traffic going over that connection.

49.Troubleshooting: Seeing error, "Could not update Product Names":

If you see this error:

Date / Time: 10/28/2004 1:40:56 PM
Task: UPDATE_PRODUCT_NAMES
Host: LocalHost
Message: Could not update Product Names: Object reference not set to an instance of an object.

To fix this, perform the following:

1. Stop the eEye Shared Services service
2. Delete the file:C:\Program Files\Common Files\eEye Digital Security\Shared Services Host\data\Discovery.xml
3. Restart Shared Services



50.Troubleshooting: communication between the agents and the Security Console:

The communication path uses:

1. Seccomm.dll is used by the runtime system to implement inter-machine communication. Most of this communication is on behalf of the console user and uses NTLM-based authentication.
2. Seccomm.dll is also used by remote deployment to communicate status back to the originating console machine. Certificates are specifically generated by DeploySupport for use in this communication.

Debug Info:

The release version of seccomm.dll does not generate debug information.

The debug version uses OutputDebugString to generate messages that are caught by a debugger or DebugView (better).

1. to generate debug info in the runtime system the debug version of seccomm.dll must be placed in the system32 directory and AppBus restarted. DebugView can then be used to capture the trace data.
2. to generate debug info as part of remote deployment the debug version must be placed in the Program Files\Common Files\eEye Digital Security\Shared Services Host\data\remoteservice prior to executing the remote deployment job. When the file debug_SyncIt.log is found in the current directory then debug information is appended to that file. This is the same information as that in 1.

Seccomm provides the negotiation of authenticated and encrypted communication channels. It can provide information about the authentication process and also can be used to view the data being communicated in an unencrypted format (as opposed to say a line trace).

Though tracing the data can be useful, the primary use of seccomm debug data is probably to debug failed authentication processing, whether NTLM or certificate based. In the case of NTLM it is unlikely that more information will be gained, since the system generated error code is usually surfaced to the application bus. In the case of certificates it seems that there are error situations that can arise and the debug info can sometimes be useful in troubleshooting this situation.

51.Troubleshooting: Blink Wireless debugging:

Try turning off the Enable API protection. To date we know that this allows UltraEdit Wireless adapters to work.

52.Troubleshooting: Blink Script Engine debugging:

Failures will appear in the windows event log or if the script is being executed through the command line the error will be outputted on the command screen.



53.Troubleshooting: “Could not update Product Names”:

Date / Time: 10/28/2004 1:40:56 PM
Task: UPDATE_PRODUCT_NAMES
Host: LocalHost
Message: Could not update Product Names: Object reference not set to an instance of an object.

To fix, do the following:

4. Stop the eEye Shared Services service
5. Delete the file:C:\Program Files\Common Files\eEye Digital Security\Shared Services Host\data\Discovery.xml
6. Restart Shared Services

54.Troubleshooting: SyncIt – Blink product updating:

- The updater creates a log file called debug_syncIt.log in C:\Program Files\Common Files\eEye Digital Security\SyncIt.
- In case of an interactive run of the updater, it will generate an xml
- A dmp file will be created if there is a crash.

55.Troubleshooting: Blink asked for the license key twice because it’s expired:

- From the Blink Console you created an automated, silent, package WITH a license key.
- You deployed the package to another PC and then the system requested a reboot.
- Upon logging in again, Blink started and prompted for a license, but this was supplied during the package creation but never passed though!
- The problem is that the license has expired.

56.Troubleshooting: Blink I/O “The root element is missing”:

The root element is missing is a 1.0 incompatibility with later versions. Perform the following:

- Uninstall the security console.
- Delete the subdirectory containing the files for the console.



- Re-install the security console.

57.Troubleshooting: Can't get past the Admin credentials prompts:

In the security settings for XP, a security option had to be changed. Network Access: Sharing has to be changed to Classic from Guest

58.Troubleshooting: How to enable Application Protection in Blink 1.X Versions:

Blink V2.0 will have a User Interface for Application Protection

To enable Kevlar set this registry key before starting Blink

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SOFTWARE\eEye\Blink]
"KevlarEnabled"=dword:00000001
```

59.Troubleshooting: How remove Blink and Security Console Manually:

To remove manually you need to remove these items:

1. The main install directory
2. The eEye Digital Security folder under Common Files. This directory is shared with other eEye products so removing it will break any eEye products on the machine.
3. The main eEye registry key under HKEY_LOCAL_MACHINE. This key is shared with other eEye products.
4. The main eEye Digital Security registry key under HKEY_CURRENT_USER. This key is shared with other eEye products.
5. To remove the MSI references you can use a tool called msizap.exe from Microsoft. This is a command line tool. Or there is a GUI tool called msicuu.exe. (MSI Cleanup utility). The GUI tool lets you select the product from a list. It then calls msizap.exe that comes with the tool. The version of msizap.exe that comes with the GUI utility is for the old version of MSI. Our recommendation is to download msicuu.exe from Microsoft. Then replace the included msizap.exe with the latest version of the utility (which can be found in Platform SDK). Then run the MSICUU utility. If you want to skip the MSICUU, you can run msizap.exe from the command line, but you must pass it the Product ID (GUID).

60.Troubleshooting: Blink to Security Console Communication Requirement:

The Blink communications channel, named, "Application Bus" that performs console authenticating to the remote blink host utilizes NTLM version 1 Authentication. Hosts set to only accept NTLM ver 2 authentication requests will cause the communication to fail.

The Blink enabled host must be set to accept NTLM Version 1 Authentication requests for the Security Console to successfully connect.



61.Troubleshooting: Security Console Discovery View:

1. Stop eEye Shared Services service
2. Edit C:\Program Files\Common Files\eEye Digital Security\Shared Services Host\eyessh.exe.config and replace <add name="TraceLevelSwitch" value="0" /> with <add name="TraceLevelSwitch" value="4" />
3. Start the shared services service.
4. Review or insert into a ticket, the file C:\Program Files\Common Files\eEye Digital Security\Shared Services Host\SharedServicesTraceLog.txt

62.Troubleshooting: How to test Security Console to Blink connectivity:

From the Security Console command line (DOS) interface type "telnet <Blink remote IP> 2000". Telenet should connect and present a blank screen. If it fails to connect, you will see an error message in the I/O.

63.Troubleshooting: How to "Import Hosts" into the Security Console :

There are more extensive guidelines for creating an import file for the Security Console Discovery

It's strongly recommended that at least the IP and the ComputerName should be provided per asset.

Examples of how crate Import Hosts files:

By IP Addresses only:

192.168.0.99
192.168.0.100

By host name only:

HostA
HostB



By IP Address, ComputerName and the DnsName:

192.168.0.100, TEST, test.mydomain.com

64. Troubleshooting: How to test Security Console to Blink deployment:

- *Recheck that the password that you entered for remote deployment is correct.*
- *To test the file transfer channel, open the share on the target host: \\remote_ip\admin\$ and copy a file into the Temp folder.*
- *A folder named RDLogs will be created in installation folder of the target agent (c:\program files\eEye Digital Security\Blink) that contains 4 log files: debug_syncIt.log, application bus installer log, blink installer log and eeyeremoteinstall service log*
- *. If this directory doesn't exist then search for these logs in %Temp% \eEye Digital Security\RDLogs.*

65. Troubleshooting: How to save discovery data:

- *In the Security Console, you can export the Network View data to a csv file and re-import it later.*



Appendix A

Application Protection Configuration

Configuring settings for a specific application

Open the apex.ini file in the Config folder. (inside Blink's installation folder)

Add a line with this syntax
process;MD5;Kevlar;action

Process

This field can be any of the following:

The process name. Ex. Iexplore.exe

The full path to the process. Ex "c:\Program Files\Internet Explorer\iexplore.exe"

The universal path to the process. Ex: "%ProgramFiles%\Internet Explorer\iexplore.exe"

* (star). If this is used, the rule will be applied to all processes!

MD5

This is the MD5 checksum of the binary image of the process.

Ex: 0F7D9C87B0CE1FA520473119752C6F79

Note: If the MD5 field is present, the Process field will be ignored

Kevlar

This has to be set to the string "Kevlar"

Action

Can be any of the following:

0 – Ignore the alert for this process. Incident will be logged.

1 – Reserved code. Do not use.

2 – Terminate the offending thread. This option is suitable for servers with multiple threads serving client requests. Killing the exploited thread will leave the server operational.

3 – Terminate the exploited process.

4 – Simply deny the malicious call.

5 – Terminate the process and restart it. This should be the preferred option. The process will be restarted with the same command line and under the same account as the original process.

-1 – Ignore the alert and do not create a log entry about it

AP employs a number of protection techniques which can be independently turned ON and OFF through the use of the registry key, HKEY_LOCAL_MACHINE\SOFTWARE\eEye\Blink .

To change the default behavior of AP, create a DWORD value named KevlarOptions and assign it a combination of the following flags:



```
KEVLAR_PROT_PEB = 0x0000001,  
KEVLAR_PROT_UNHANDLED_EXCEPT = 0x00000002,  
KEVLAR_PROT_CREATE_FAKE_HEADERS = 0x00000004,  
KEVLAR_PROT_SANITIZE_REGISTERS = 0x00000008,  
KEVLAR_PROT_MOVE_KERNEL32_POS = 0x00000010,  
KEVLAR_PROT_CMFLAG_JITCOMPAT = 0x40000000,  
KEVLAR_PROT_CMFLAG_PACKERCOMPAT = 0x20000000,  
KEVLAR_PROT_CMFLAG_DEBUG = 0x00010000,  
KEVLAR_PROT_CMFLAG_WPACOMPAT = 0x10000000
```

If a flag is set, that specific level of protection will be enabled.

To compute the desired value, open Calc.exe, switch it in Scientific mode, select Hex mode and do the following operation: flag1 OR flag2 OR flag3 etc

Shortcut:

To enable debugging, set the following value: 0x7001001E

Example: To set the KEVLAR_PROT_PEB and KEVLAR_PROT_CMFLAG_DEBUG flags, use this value 0x00010001

Flag Definitions:

```
KEVLAR_PROT_PEB = 0x00000001,
```

Checks the PEB locking function pointers (common targets for arbitrary-memory-overwrite exploits) whenever an exception occurs, to ensure that they point to the proper functions (NTDLL!RtlEnterCriticalSection and RtlLeaveCriticalSection).

```
KEVLAR_PROT_UNHANDLED_EXCEPT = 0x00000002,
```

Checks the Unhandled Exception Filter function pointer in KERNEL32.DLL, another common target for arbitrary-memory-overwrite exploits, to make sure that it is only modified by calling KERNEL32!SetUnhandledExceptionFilter (which is the orthodox way to change this function pointer).

```
KEVLAR_PROT_CREATE_FAKE_HEADERS = 0x00000004,
```

Makes a "dummy" copy of the MZ/PE executable headers within a loaded module's unused header space. This countermeasure may cause some exploit payload's "RVA loaders" to crash before the operational part of the payload can be executed, if they use techniques that involve scanning backwards through a module for the DOS header's "MZ" signature.

```
KEVLAR_PROT_SANITIZE_REGISTERS = 0x00000008,
```



Causes Structured Exception Handling routines to be called with all useful values purged from the registers, rendering traditional SEH-based exploitation (e.g., CodeRed's "CALL EBX") non-functional and prone to crash instead. Windows XP already offers this type of sanitization.

KEVLAR_PROT_MOVE_KERNEL32_POS = 0x00000010,

Rearranges the first two nodes in the three doubly-linked module lists, which will cause payloads using LSD's "RVA technique" (common across most modern payloads) to retrieve the image base for either the application module or NTDLL.DLL, either way resulting in a crash rather than successful payload execution.

KEVLAR_PROT_CMFLAG_DEBUG = 0x00010000,

Enables Kevlar debugging output to debuggers and to log files in the "C:\k20debug" directory.

KEVLAR_PROT_CMFLAG_WPACOMPAT = 0x10000000

Enables compatibility with Windows Product Activation (Windows XP and later) by automatically weakening protections within a process if LICDLL.DLL is loaded. Protections are loosened by applying CMFLAG_ALLOWWX for the duration of the process's execution (see below).

KEVLAR_PROT_CMFLAG_PACKERCOMPAT = 0x20000000,

Enables compatibility with some common executable packers (e.g., ASPack) which inappropriately leave unpacked image code without executable memory permissions after loading. If this flag is set, code will be allowed to execute in SEC_IMAGE memory with any set of page permissions, if the section is 1) flagged as containing code, 2) flagged as requesting executable permissions, or 3) named ".text" or ".aspack" (this list may be modified in the future), according to the section table in the module's headers.

KEVLAR_PROT_CMFLAG_JITCOMPAT = 0x40000000,

Enables compatibility with applications containing .Net managed code that perform API calls using Just-In-Time compiled code. If a module is loaded that appears to contain managed code (the IMAGE_DIRECTORY_ENTRY_COM_DESCRIPTOR data directory entry is non-empty), then protections will be weakened by applying CMFLAG_ALLOWWX for the remainder of the process's existence.

// CMFLAG_ALLOWWX = 0x80000000



Allows code execution within PAGE_EXECUTE_READWRITE VirtualAlloc'ed memory, as used by Windows Product Activation (LICDLL.DLL and WINLOGON.EXE) and JIT-compiled code (.Net managed code, some Java run-times). Normally code is not allowed to execute in writable VirtualAlloc'ed memory (as opposed to SEC_IMAGE memory, which is part of a loaded module and is allowed to execute code from within writable memory).