

eEye® Digital Security

# ***V2.0 FAQ for Blink***

**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:
  - **Scan for Malware**
  - **Turn on Application Protection and put your host into “the wild”**
  - **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?**

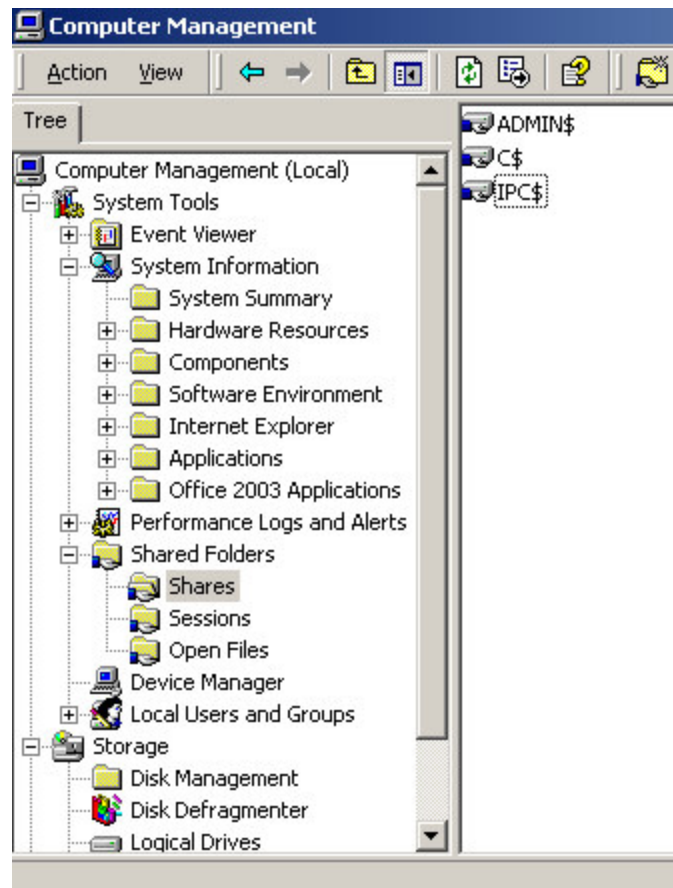
- The History and Queue Views in the Security Console show what it executed
- Each target host has the directory C:\Program Files\Common Files\eEye Digital Security\Shared Services Host\data\Packages that contains a history of each package and the detailed settings of that deployment.

**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

```
C:\WINNT\system32\cmd.exe
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\Documents and Settings\jelliott>net share

Share name      Resource                Remark
-----
ADMIN$          C:\WINNT                Remote Admin
C$              C:\                     Default share
IPC$            \*.*                   Remote IPC
The command completed successfully.

C:\Documents and Settings\jelliott>_
```

***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.



## Using Policy Enforcement to Enforce Security Standards

---

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]

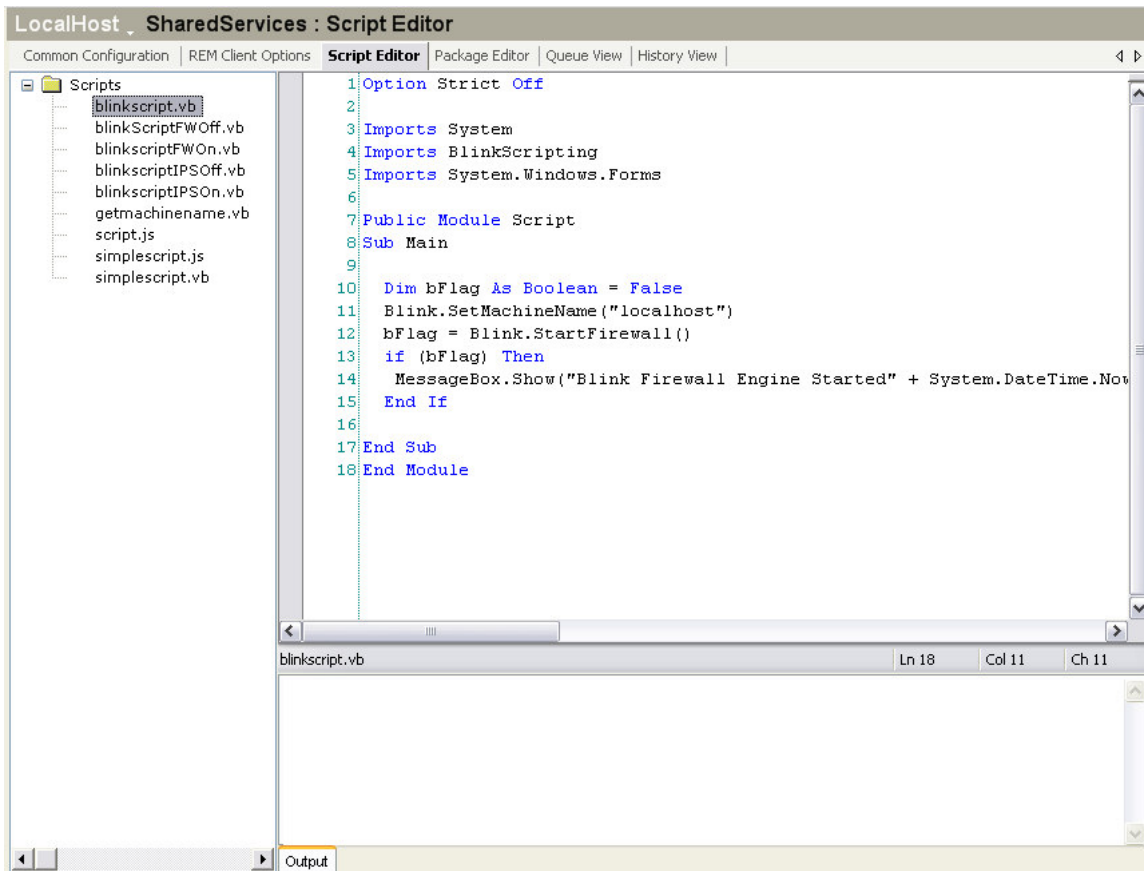
"ForceAllEventsLogging"=dword:00000001

### **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 manor 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.*



## Using Policy Enforcement to Enforce Security Standards

---

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.