Over the last few years the volume of malware has grown dramatically, thanks mostly to automation and exploit kits that make it easier for cybercriminals to create and distribute it. Kits are an intrinsic part of crimeware, providing not only the tools for creation and distribution of malware, but also the systems to manage networks of infected computers.

This paper provides insight into two of the most commonly used and technically capable pieces of crimeware, the Blackhole exploit kit and the ZeroAccess rootkit. We explain why these kits are so useful to the bad guys and show you how you can stop these threats from infecting your network and your users.
What is Blackhole?

Blackhole is what is known as an exploit kit. An exploit kit is a tool used by attackers to get their software installed on a victim’s PC. As the name suggests, these kits exploit security holes in the installed software to deliver their chosen, and almost always, malicious payloads.

Surprisingly, the people who write Blackhole aren’t the ones who use it. They aren’t interested in sending spam or deploying botnets or other malware. Their business is purely to create and sell exploit kits as a service to other cybercriminals.

Exploit writers deliver, package and sell their kits much like SaaS (software as a service) products. As with similar kits, a cybercriminal can license Blackhole for a period of three, six or twelve months. Varying in costs and options, some kits include updates to the exploits over the course of a license, while others are available at a premium to include more recent exploits, zero-days (previously unknown exploits) or versions that can get around anti-malware software.

An analysis of detected web threats between October 2011 and March 2012 found that almost 30% of the detected threats were either Blackhole or redirects from compromised legitimate sites to Blackhole kits, illustrating just how prevalent this form of attack is.
How Blackhole infects your users

As with all attacks using exploit kits, attackers need to direct a user’s web browser to an exploit site. Blackhole uses the following techniques to send users to an exploit site.

**Compromised webpages:** The attackers compromise legitimate websites or servers to serve their malicious code. When users browse these pages, the malicious code silently loads content from the exploit site. Often the injected redirects do not link directly to the Blackhole exploit site. Instead they reference a remote server that bounces the request to an exploit site. This approach allows exploit writer to sell user traffic as a commodity. Links to these pages can also spread by other methods such as enticing messages on Twitter. For example links spread via Twitter point to Russian webpages that attempt to infect your Windows PC with Blackhole.¹

**Spam messages:** Despite years of user education warning of the dangers of links or attachments in email messages, spam continues to be a useful tool for attackers to trick users. Cybercriminals may send spam including a simple URL link within the email message, which loads JavaScript content to redirect to the Blackhole site or an email message containing a HTML attachment.

**Landing page:** Whatever method is used to control user web traffic, the result is the same: the user’s browser loads code served up from the “landing page” of the exploit kit. The landing page:

- Credits page requests to specific individuals or groups responsible for redirecting the victim for payment purposes
- Fingerprint the machine to identify OS, browser and browser version, Adobe Flash, Adobe Reader versions, Java version
- Loads the relevant exploit components (PDF, Flash, Java—the file types used by Blackhole)

---

How a Blackhole attack works

The first component of Blackhole is a heavily obfuscated JavaScript. Webpages on compromised sites typically contain malicious JavaScript since it provides more opportunities for the attackers to hide the malicious code. In fact, aggressive code obfuscation makes Blackhole one of the most persistent threat campaigns experienced to date.

The script seeks out the weaknesses in the PC's defenses. Typically it will use multiple different types of encoding in an attempt to make analysis and detection by anti-malware more difficult. The script's purpose is simple. It determines which browser and plugins are available and which versions are installed. Blackhole has a variety of exploits available to use when attacking and will select those that will be effective against the installed software.

For example, an attack attempting to load a Flash animation might check the version of Adobe Reader and supply an older exploit for an older version or newer exploit for a newer version. It might try an attack against Windows Help and Support Center and send a version-appropriate attack against Java. Once the code has assessed the defenses it will try whichever attacks are likely to succeed.

All of these attacks have the same goal: to run a piece of malware on the target computer. The attackers who have licensed or rented Blackhole decide what type of malware Blackhole will deliver. Criminals using Blackhole will commonly choose to deliver threats such as fake antivirus, ransomware (which encrypts the user's files and demands a ransom to release them), a data-stealing Trojan such as Zeus, or rootkits such as TDSS or ZeroAccess.

How to protect against Blackhole

Generally speaking there are three stages of a Blackhole attack:

1. The initial contact either by email or compromised website
2. Redirection to the attack site which probes for weaknesses
3. Delivery of the exploit itself and the resulting malware drop

Fortunately, you can protect against Blackhole attacks using security technologies and methods.

**Spam filters:** You can intercept the initial email contact using a spam filter with content scanning, which blocks the email by detecting either the obfuscated JavaScript content or its botnet-based origination.

**Web filters:** A web filter can block the web components of an attack, either at the network or endpoint level, by actively scanning content to identify the probing JavaScript. However, a simple reputation filter would fail to block attacks using a compromised legitimate website.
Patching: Patching your operating systems and applications provides an additional defense against Blackhole and other exploits. While patching cannot prevent zero-day vulnerabilities, Blackhole and other similar kits mainly target application vulnerabilities in Flash, Adobe Reader and Java. You can protect these applications by turning on functionality that either automatically updates the software or reminds you when updates become available.

What is ZeroAccess?

ZeroAccess is a family of rootkits and backdoors criminals use to install and conceal other malware, commonly for redirecting a user’s web traffic. It uses rootkit techniques to hide from security software while allowing remote attackers to control infected computers. The concealment of a rootkit makes it more likely that an attack will remain unnoticed, allowing criminals to steal more information and take advantage of a compromised network for a long time.

Like Blackhole, ZeroAccess is a link in the attack chain. However, ZeroAccess is designed to conceal its own presence in order to secretly download and install other malware. Because it’s hard to detect, malware distributors may prefer to use ZeroAccess rather than simply spreading the final stages of their attacks directly.

Similar to other botnets, cybercriminals rent access to the ZeroAccess network to send spam, commit fraud and perform other malicious activities.

How ZeroAccess infects your users

ZeroAccess infects users through vectors similar to other high-profile malware families currently circulating. The main distribution methods can be split into two categories: Exploit kits and social engineering.

Exploit kits

ZeroAccess is a popular payload for various exploit kits currently on the market, including Blackhole. An exploit pack typically comes as a series of PHP scripts that are stored on a web server under the control of the attacker. When a victim’s browser accesses an infected website, the server backend attempts to exploit a vulnerability on the target machine and execute the payload. Exploit packs usually contain many different exploits targeting applications commonly found on Windows PCs such as Internet Explorer, Acrobat, Flash and Java.

Attackers commonly drive traffic to websites hosting exploit packs through legitimate sites compromised through stolen FTP credentials or SQL injection. These compromised sites host the exploit packs themselves and redirect traffic to send the user to the attack site.
Ad servers have also been compromised in this way to spread the infection to a number of sites hosting the ads. Ad servers can spread an infection very quickly if the ads appear on high profile websites. SEO (search engine optimization) techniques drive compromised websites up search engine rankings, increasing traffic to the attack site.

We have also seen attacks initiated through email. Attackers send out spam containing a link that, when clicked, sends the victim to a compromised website hosting an exploit pack.

Exploit packs as an infection vector for ZeroAccess usually require no input from the victim other than browsing to an apparently legitimate website or clicking an innocuous-seeming link.

**Social engineering**

The second main infection vector for ZeroAccess is through a variety of social engineering techniques. At the heart of these is the goal of convincing a victim into running an executable that they should not. The lure is often a piece of illicit software such as a game or a copyright protection bypassing tool such as a crack or keygen. These Trojanized files are placed on upload sites and on torrents and given filenames designed to trick the unwary into downloading and running them.

**How a ZeroAccess attack works**

**Dropper**

The first stage of a ZeroAccess attack is a dropper that installs the ZeroAccess components on the target computer.

ZeroAccess droppers have changed as the rootkit itself has evolved. Currently, droppers are usually obfuscated using complex polymorphic packers, software designed to modify the appearance of a program so that no two instances look the same to a security scanner.

These packers are a typical example of the protection measures that modern malware employs to both hinder analysis and to avoid detection by security tools. They add layers of compression and encryption to a program to conceal its true purpose. They are updated several times a day and are always checked against anti-malware scanners before they are released into the wild.

Packers have many anti-emulation and anti-debug techniques designed to defeat emulators inside anti-malware scanning software, making analysis inside the controlled environment of a research lab more difficult. The dropper has recently been using hardware breakpoints as part of its unpacking routine, which makes attaching a kernel debugger to the target system (necessary to analyze the kernel-mode components) more challenging.

An interesting feature of ZeroAccess droppers is that a single dropper will install the 32-bit or the 64-bit version of the malware depending on which version of Windows is running.
Payload

Computers infected with ZeroAccess communicate with each other and form a peer-to-peer botnet. The payload of ZeroAccess connects to this botnet and downloads further files. The network communication initiates both from the kernel driver itself and from a component injected by the driver into user memory, usually inside either the address space of explorer.exe or svchost.exe. Injecting its code into user processes such as explorer.exe and svchost.exe allows ZeroAccess to bypass some client firewalls.

When initially installed, ZeroAccess includes a file that contains a list of 256 IP addresses of previously infected computers. Each IP address is followed by a 32-bit time value that probably indicates the last contact time for each IP address as the list is sorted by the time value, highest first. This is the initial list of peers that the infected machine knows about in the botnet. The bot will attempt to contact each IP address in the list on a fixed port number that is stored inside the bot executable file. Once a successful connection is made, ZeroAccess receives commands from the botnet.

The other main payload is a spambot. When this payload is downloaded it installs itself, downloads spam templates and target email addresses, and sends spam. It is likely that the authors of the spambot are renting a portion of the ZeroAccess botnet to deliver their malware.

Hidden ZeroAccess Files

ZeroAccess attempts to make its files difficult to access by hiding inside the Global Assembly Cache (GAC).
Technical features of ZeroAccess

Apart from the stealthiness that ZeroAccess gets from its rootkit origins, two other features make it successful.

Some versions of ZeroAccess use aggressive techniques to defend themselves on infected endpoints. It’s common for malware to attempt to disable security software. Usually the malware has a list of security programs it will attempt to kill if they are running on target computers. However, that method won’t work against software that implements some randomness in its file and process names, a common technique in anti-rootkit software.

To counter this anti-rootkit defense, ZeroAccess creates a dummy or trap process which acts as bait for security software. ZeroAccess monitors programs that attempt to access the dummy process, which it assumes to be a security scanner. ZeroAccess then tries to disable the scanner by terminating its running processes and changing its access permissions so that it cannot be run again. However, this kind of damage to security software may reveal the presence of a rootkit. More recent versions of ZeroAccess don’t use this technique.

The second aspect of ZeroAccess that makes it resilient lies in the organization of its botnet infrastructure.

ZeroAccess operates as a botnet, meaning that to be useful it must have some way to receive commands. For many botnets the command and control infrastructure is their weakness. Remove the key command and control servers and the individual PCs are left without instructions and the botnet becomes useless to criminals.

To avoid this weakness ZeroAccess, and some other recent botnets, employ a distributed or peer-to-peer control model. Individual nodes can be cleaned up and removed from the network but it cannot be killed at a single stroke.

However, this botnet model has some weaknesses too. The individual nodes of the botnet have to know of other nearby nodes in order to receive instructions—and those instructions may take time to propagate. Also, nodes that do not have direct Internet access cannot act as servers for nodes in other networks. To account for this, each installation of ZeroAccess contains a configuration file with addresses of 256 previous nodes to ensure that it can contact another infected computer for instructions.

For ZeroAccess, the peer-to-peer model mainly helps it to distribute other malware. It also helps ZeroAccess to support click fraud, where PCs visit a website or access online ads to generate advertising income for the affiliate serving those ads. This botnet setup also allows ZeroAccess to distributes spambots, which use the infected PCs to send spam. It’s likely that click fraudsters, spammers and malware authors rent space on the ZeroAccess botnet, funding the profits of its authors. That means the cybercriminals can invest in the continued development of ZeroAccess to make it even more stealthy and dangerous.
How to protect against ZeroAccess

As with Blackhole, much of the interesting action for ZeroAccess happens on the endpoint, so most of the useful defenses are based at the endpoint.

**Anti-rootkit tools:** For the true rootkit versions, an anti-rootkit tool or an anti-malware scanner with anti-rootkit capabilities is the best defense. Typically these will find the modifications in the operating system kernel and proceed to clean up from there. Due to the nature of cleaning an infection from kernel memory, a system restart can be required. So keep an eye on cleanup logs to ensure that the removal is being completed.

**Logs:** Monitoring security console or management logs for errors can tip you off to a network incursion. Remember that some versions of ZeroAccess, and other malware, will attempt to disable installed security products. So a sudden rash of failure reports could be related to an infection.

**Firewalls:** Perimeter or client firewalls can disrupt the peer-to-peer communications of a botnet such as ZeroAccess, although it gets some information via HTTP, which is unlikely to be blocked. Much of the P2P communication is done at high port numbers not used by common services.

ZeroAccess has adapted as its target environment has evolved, adding compatibility for 64-bit architectures and multi-user, multi-privilege systems. As a result, ZeroAccess should be considered an advanced and dangerous threat that requires a multi-layered protection strategy.
Sophos EndUser Protection

Endpoint protection gets personal. It’s personal because it’s all about protecting the end user and not just a laptop or a mobile device. You get everything you need to protect your users, on every device, everywhere they go. And it’s not only personal, it’s easier to buy, deploy, and manage too.

Sophos EndUser Protection gives you endpoint security and mobile device management all in one

- Secures Windows, Mac, Linux, iOS, Android, and more
- Provides proven protection against malware
- Lets user securely store and share data
- Makes web browsing safe with built-in URL filtering
- Simplifies management with workflows engineered for business
- Reduces complexity with a single license from a single vendor