SOPHOS
Security made simple.

Exploits Explained:

Comprehensive Exploit Prevention and Vendor Offerings

Exploits take advantage of weaknesses in legitimate software products like Adobe Flash and Microsoft Office to infect computers for criminal purposes. They're commonly leveraged by cybercriminals in order to penetrate organizations' defenses. The objectives of these criminals are diverse: stealing data or holding it for ransom, performing reconnaissance, or simply as a means to deploy more traditional malware.

It's common to find exploits used as part of cyber attacks: upwards of 90% of reported data breaches find that an exploit is used at one or more points in the attack chain. Including exploit prevention as part of a comprehensive lineup of security defenses is clearly valuable.

Exploits have been around for more than 30 years, so it should come as no surprise that almost every major security vendor can claim some level of exploit prevention. However, the breadth and depth of that protection varies significantly between vendors. For some, it's a box to tick; for others, it's a major focal point. Read this paper to learn more about exploits and the various levels of exploit prevention found in prominent security products.

The Exploit Industry: Crimeware as a Service

Thanks to exploit kits, malware authors don't need to worry about how to find bugs in Java or Silverlight or Flash; how to build those bugs into working exploits; how to find insecure web servers to host the exploits; or how to entice prospective victims to the booby-trapped web pages.

Likewise, exploit kit authors don't have to worry about writing full-blown malware; they don't have to run servers to keep track of infected computers or to collect money from individual victims; they don't have to get involved in the exfiltration of stolen data or selling that data.

With cybercrime now a multi-billion-dollar industry that is projected to cause nearly $2 trillion in damages by 2019, each aspect of an attack has been industrialized. Criminals have the luxury of being able to specialize in one or more parts of the threat landscape in what's become known jokingly as CaaS – or "Crimeware as a Service."

In this now-lucrative industry, exploit brokers have emerged: they buy exploits from people who discover them and sell them to people who want to make use of them, whether government agencies or nefarious hackers.

Buyers invariably keep their purposes to themselves. As Kevin Mitnick, founder of Mitnick's Absolute Zero Day Exploit Exchange, explained to Wired, "When we have a client that wants a zero-day vulnerability for whatever reason, we don't ask, and in fact they wouldn't tell us. Researchers find them, they sell them to us for X, we sell them to clients for Y and make the margin in between."

> "When we have a client that wants a zero-day vulnerability for whatever reason, we don't ask, and in fact they wouldn't tell us. Researchers find them, they sell them to us for X, we sell them to clients for Y and make the margin in between."
>
> KEVIN MITNICK

Exploit Mitigation Techniques

With more than 400,000 unique malware samples created each day and thousands of new vulnerabilities discovered each year, the challenge of preventing malicious attacks is daunting. This explosion of growth in malware variants requires new and innovative approaches when it comes to defending against cybercriminals.

A careful examination of the modern cybercrime industry shows an opportunity for asymmetric defense. As it turns out, despite the seemingly endless parade of new attacks, there are only about 20 or so techniques that can be used to exploit software. So an approach that's able to counteract this handful of exploit techniques – instead of targeting each and every exploit – is extremely powerful.

What's more: depending on the vulnerability, attackers often end up having to chain a handful of exploit techniques together to get to the stage where they can deliver malware. These techniques don't change much from year to year: perhaps one or two new tricks are added to the list of available techniques.

When evaluating major security products, the absence of significant exploit technique mitigation can be surprising. And while some of the newer vendors who claim to offer next-generation technology have broader support for exploit mitigation, even here the coverage is spotty.

Below is a list of 23 exploit techniques that are used by cybercriminals and nation-states. Mitigations for each technique will vary by vendor. It is important to know that when a vendor claims to prevent exploits, most vendors simply protect against a fraction of the commonly used exploit methods. Only Sophos provides truly comprehensive exploit prevention.

Enforce Data Execution Prevention (DEP)

Data execution prevention (DEP) is a set of hardware and software technologies that perform additional checks on memory to help prevent buffer overflows. Without DEP, an attacker can attempt to exploit a software vulnerability by jumping to malicious code (shellcode) at a memory location where attacker-controlled data resides, such as the heap or stack. Without DEP, these regions are normally marked as executable, so malicious code will be able to run.

DEP is an opt-in option for Windows XP and above that must be set by the software vendor when building an application. Furthermore, attacks are available for bypassing built-in DEP protection and, as such, dependence on the operating system implementation is not recommended.

Vendors mitigating this exploit technique: Sophos Intercept X, Microsoft EMET, Malware Bytes Anti-Exploit, Palo Alto Network Traps, Crowdstrike Falcon

Mandatory Address Space Layout Randomization (ASLR)

Some exploits work by targeting memory locations known to be associated with particular processes. In older versions of Windows (including Windows XP), core processes tended to be loaded into predictable memory locations upon system startup. Address space layout randomization (ASLR) randomizes the memory locations used by system files and other programs, making it much harder for an attacker to correctly guess the location of a given process.

ASLR is only available on Windows Vista and above and, like DEP, must be set by the software vendor when building an application. And like DEP, attacks are available for bypassing built-in ASLR protection and, as such, dependence on the operating system implementation is not recommended.

Vendors mitigating this exploit technique: Sophos Intercept X, Microsoft EMET, Palo Alto Networks Traps, Crowdstrike Falcon

Bottom-up ASLR

If enabled, the mandatory ASLR found on a Windows machine only changes the base address of applications once and then persists until the machine is rebooted. Attackers can take advantage of this in order to enable the re-use of discovered locations for applications that are started multiple times.

Bottom-up ASLR improves the entropy or randomness of mandatory ASLR. The main advantage of bottom-up ASLR is that it changes the base address of protected applications each time the application is started.

Vendors mitigating this exploit technique: Sophos Intercept X, Microsoft EMET, Malwarebytes Anti-Exploit

Null Page (Null Dereference Protection)

Starting with Windows 8 and onwards, Microsoft denies programs the ability to allocate and/or map the "NULL page" (memory residing at virtual address 0x00000000 in the address space). By doing this, Microsoft successfully mitigates the direct exploitation of a whole class of vulnerabilities called "NULL pointer dereference" vulnerabilities.

On Windows XP, Windows Vista, and Windows 7, the exploitation of such a flaw would allow the attacker to execute code in the context of the kernel (under the ring0 CPU privilege level), resulting in privilege escalation to one of the highest levels. Such vulnerabilities give attackers access to virtually all parts of the operating system.

Vendors supporting this mitigation technique: Sophos Intercept X, Microsoft EMET

Heap Spray Allocation

A heap spray is a technique that does not actually exploit vulnerabilities but is used to make a vulnerability easier to exploit. Using a technique called Heap

Feng Shui1 an attacker is able to reliably position intended data structures or shellcode on the heap, thus facilitating a reliable exploitation of a software vulnerability.

Vendors supporting this mitigation technique: Sophos Intercept X, Microsoft EMET, Palo Alto Networks Traps, Crowdstrike Falcon

Dynamic Heap Spray

The dynamic heap spray mitigation analyzes the contents of memory allocations to detect patterns that indicate heap sprays containing NOP sleds, polymorphic NOP sleds, JavaScript arrays, ActionScript arrays, and other suspicious sequences that are placed to facilitate exploit attacks.

Vendors supporting this mitigation technique: Sophos Intercept X, Palo Alto Networks Traps

1 https://cansecwest.com/slides/2014/The%20Art%20of%20Leaks%20-%20read%20version%20-%20Yoyo.pdf

Stack Pivot

The stack of an application is a memory area that contains, among other things, a list of memory address locations (so-called return addresses). These locations contain the actual code that the processor needs to execute in the near future.

Stack pivoting is widely used by vulnerability exploits to bypass protections like DEP, for example by chaining ROP gadgets in a return-oriented programming attack. With stack pivoting, attacks can pivot from the real stack to a new fake stack, which could be an attacker-controlled buffer such as the heap, from which attackers can control the future flow of program execution.

Vendors supporting this mitigation technique: Sophos Intercept X, Cylance PROTECT, Microsoft EMET, Malwarebytes Anti-Exploit, Palo Alto Networks Traps

Stack Exec (MemProt)

Under normal circumstances, the stack contains data and addresses pointing to code for the processor to execute in the near future. Using a stack buffer overflow2, it is possible for attackers to overwrite the stack with arbitrary code. In order to make this code run on the processor, the memory area of the stack must be made executable to circumvent DEP. Once the stack-memory is executable, it is very easy for an attacker to supply and run program code.

Vendors supporting this mitigation technique: Sophos Intercept X, Cylance PROTECT, Microsoft EMET

Stack-based ROP Mitigations (Caller)

Control-flow integrity (CFI) technology is an approach to prevent attackers from hijacking control-flow of internet-facing applications like web browsers, Microsoft Office, and other productivity and media software. To defeat security technologies like data execution prevention (DEP) and address space layout randomization (ASLR), control-flow attacks are now common practice. These attacks are invisible to antivirus, most "next-gen" products, and other cyber defenses as there are no malicious files involved. Instead, the attack is constructed at run time by combining short pieces of benign code that are part of existing applications like Internet Explorer and Adobe Flash Player – a so-called code-reuse or return-oriented programming (ROP) attack.

Vendors supporting this mitigation technique: Sophos Intercept X, Kaspersky Endpoint Security, McAfee Endpoint Security, Microsoft EMET, Malwarebytes Anti-Exploit, Palo Alto Networks Traps

2 https://en.wikipedia.org/wiki/Stack_buffer_overflow

Branch-based ROP Mitigations (Hardware Augmented)

ROP attacks can be achieved by leveraging an unused hardware feature in mainstream Intel® processors (from 2008 and newer) to track code execution and augment the analysis and detection of advanced exploit attacks at run time. Employing read-only hardware-traced (branch) records has a significant security benefit over software stack-based approaches. The branch information that can be retrieved from these records not only identifies the target of the branch, but also the source. So it actually shows where the change in control-flow originated from. This specific information cannot be obtained with the same level of confidence using a stack-based solution.

Branch information in the hardware-traced records cannot be manipulated; there's no way for it to be overwritten with controlled data by an attacker. Stack-based solutions (like Microsoft EMET and Palo Alto Networks Traps) rely on stack data, which is – especially in case of a ROP attack – under control of the attacker, who in turn can mislead the defender. In contrast, the hardware-traced data examined by Sophos Intercept X is more reliable and tamper resistant.

Sophos Intercept X will automatically employ Intel MSR hardware registers when it detects an Intel® Core™ i3, i5, or i7 processor (CPU). If the endpoint does not have a supported processor, Sophos Intercept X will automatically fall-back on software-only stack-based control-flow integrity checks

Vendors supporting this mitigation technique: Sophos Intercept X

Structured Exception Handler Overwrite Protection (SEHOP)

An attacker can overwrite, with a controlled value, the handler pointer of an exception record on the stack. Once an exception happens, the operating system will walk the exception record chain and call all the handlers on each exception record. Since the attacker controls one of the records, the operating system will jump to wherever the attacker wants, giving the attacker control over the flow of execution.

SEHOP is an opt-in option on Windows Vista and above and must be set by the software vendor when building the application. Attacks are available for bypassing built-in SEHOP protection and, as such, dependence on the operating system implementation is not recommended.

Vendors supporting this mitigation technique: Sophos Intercept X, Symantec Endpoint Protection, Microsoft EMET

Import Address Table Access Filtering (IAF)

An attacker eventually needs the addresses of specific system functions (e.g. kernel32!VirtualProtect) to be able to perform malicious activities. These addresses can be retrieved from different sources, one of which is the import address table (IAT) of a loaded module. The IAT is used as a lookup table when an application calls a function in a different module. Because a compiled program cannot know the memory location of the libraries it depends upon, an indirect jump is required whenever an API call is made. As the dynamic linker loads modules and joins them together, it writes actual addresses into the IAT slots so that they point to the memory locations of the corresponding library functions.

Vendors supporting this mitigation technique: Sophos Intercept X, Microsoft EMET

Load Library

Attackers can attempt to load malicious libraries by placing them on UNC paths. Monitoring of all calls to the LoadLibrary API can be used to prevent this type of library loading.

Vendors supporting this mitigation technique: Sophos Intercept X, Microsoft EMET, Malwarebytes Anti-Exploit, Palo Alto Networks Traps.

Reflective DLL Injection

Normally when you load a DLL in Windows, you call the API function LoadLibrary. LoadLibrary takes the file path of a DLL as input and loads it into memory.

Reflective DLL loading refers to loading a DLL from memory rather than from disk. Windows doesn't have a LoadLibrary function that supports this, so to get this functionality you have to write your own. One benefit to writing your own function is that you can omit some of the things Windows normally does, such as registering the DLL as a loaded module in the process, which makes the reflective loader sneakier when being investigated. Meterpreter is an example of a tool that uses reflective loading to hide itself. Mitigation is performed by analyzing if a DLL is reflectively loaded inside memory.

Vendors supporting this mitigation technique: Sophos Intercept X, Palo Alto Networks Traps

Shellcode

A shell code is a piece of code used as the payload in an exploit. The shell code will start a command shell that the attacker controls. Delivery and execution of the shell code can take many forms, and detecting the adversarial deployment of shell code involves multiple techniques to address things like fragmented shell code, encrypted payloads, and null free encoding.

Vendors supporting this mitigation technique: Sophos Intercept X

VBScript God Mode

On Windows, VBScript can be used in browsers or the local shell. When used in the browser, the abilities of VBScript are restricted for security reasons. This restriction is controlled by the safemode flag. If this flag is modified, VBScript in HTML can do everything as though it's in the local shell. Consequently, attackers can easily write malicious code in VBScript. Manipulating the safemode flag on VBScript in the web browser is known as God Mode3.

As an example, an attacker can modify the safemode flag value by leveraging the CVE-2014-6332 vulnerability4, a bug caused by improper handling while resizing an array in the Internet Explorer VBScript engine. In God Mode, arbitrary code written in VBScript can break out of the browser sandbox. Thanks to God Mode, data execution prevention (DEP), address space layout randomization (ASLR), and control-flow guard (CFG) protections are not in play.

Vendors supporting this mitigation technique: Sophos Intercept X, Microsoft EMET, Malwarebytes Anti-Exploit

WoW64

Microsoft provides backward-compatibility for 32-bit software on 64-bit editions of Windows through the "Windows on Windows" (WoW) layer. Aspects of the WoW implementation provide interesting avenues for attackers to complicate dynamic analysis, binary unpacking, and to bypass exploit mitigations.

The behavior of a 32-bit application under the WoW64 environment is different in many ways from a true 32-bit system. The ability to switch between execution modes at runtime can provide an attacker methods for exploitation, obfuscation, and anti-emulation such as:

- Additional ROP gadgets not present in 32-bit code
- Mixed execution mode payload encoders
- Execution environment features that may render mitigations less effective
- Bypassing hooks inserted by security software, only in 32-bit user space

Most endpoint protection software will only hook sensitive API functions in the 32-bit user memory space if a process is running under WoW64. If an attacker is able to switch to 64-bit mode, access is gained to unhooked 64-bit versions of the sensitive API functions that are hooked in 32-bit mode.

On 64-bit editions of Windows, Sophos Intercept X prohibits the program code from directly switching from 32-bit to 64-bit mode (e.g. using ROP), while still enabling the WoW64 layer to perform this transition.

3 https://en.wikipedia.org/wiki/Glossary_of_video_game_terms#God_mode
4 https://www.rapid7.com/db/modules/exploit/windows/browser/ms14_064_ole_code_execution

For more information about abusing WoW64, see research from Duo Security: WoW64 and So Can You5 and Mitigating Wow64 Exploit Attacks6.

Vendors supporting this mitigation technique: Sophos Intercept X

Syscall

Malicious access to critical system functions in the kernel, in an attempt to bypass hooked Windows APIs, evade sandbox analysis and most protection software.

Most endpoint security products use user-mode hooks to intercept and monitor sensitive API calls. In order to bypass these hooks, an attacker can take advantage of the fact that:

- Not all API functions are hooked; only sensitive functions
- The stubs that are used to call kernel functions are very similar; only the function index is unique

For more information about abusing syscalls, see the BreakDev.org blog entry titled Defeating Antivirus Real-time Protection From The Inside7.

Vendors supporting this mitigation technique: Sophos Intercept X

Hollow Process

Process hollowing is a technique in which a legitimate process is loaded on the system solely to act as a container for hostile code; for example, svchost.exe and explorer.exe. At launch, the legitimate code is deallocated and replaced with malicious code, after which the process starts executing the malicious code. The advantage is that this helps the process hide amongst normal processes.

Vendors supporting this mitigation technique: Sophos Intercept X, Palo Alto Networks Traps

DLL Hijacking

Due to a vulnerability commonly known as DLL hijacking, DLL spoofing, DLL preloading, or binary planting, many programs will load and execute a malicious DLL contained in the same folder as a data file opened by these programs.

Vendors supporting this mitigation technique: Sophos Intercept X, Palo Alto Networks Traps

5 https://duo.com/blog/wow64-and-so-can-you
6 https://hitmanpro.wordpress.com/2015/11/10/mitigating-wow64-exploit-attacks
7 https://breakdev.org/defeating-antivirus-real-time-protection-from-the-inside/

Application Lockdown

In the event an attacker successfully exploits and bypasses all memory and code mitigations, Sophos Intercept X limits an attacker's abilities. This feature, called Application Lockdown, aims to prevent attackers from introducing unwanted code.

Application Lockdown stops attacks that do not typically rely on software bugs in applications. Such an attack could be the use of a crafted (malicious) macro in an office document attached to a (spear) phishing email, for example. Macros in documents are potentially dangerous as they are created in the Visual Basic for Applications (VBA) programming language, which includes the ability to download and run binaries from the web and also allows the use of PowerShell and other trusted applications.

This unexpected feature (or logic-flaw exploit) offers attackers an obvious advantage as they do not need to exploit a software bug or find a way to bypass code and memory defenses in order to infect computers. They simply abuse standard functionality offered by a widely-used trusted application and only need to use social engineering to persuade the victim to open the specially crafted document.

Without the need to maintain a blacklist of folders, Sophos Intercept X will automatically terminate a protected application based on its behavior; for example, when an office application is leveraged to launch PowerShell, access the WMI, run a macro to install arbitrary code or manipulate critical system areas, Sophos Intercept X will block the malicious action – even when the attack doesn't spawn a child process.

Vendors supporting this mitigation technique: Sophos Intercept X, Malwarebytes Anti Exploit, Palo Alto Networks Traps

Java Lockdown

Java applications have access to powerful and useful tools that can be leveraged for attacks, such as the ability to write to disk and update the registry.

Vendors supporting this mitigation technique: Sophos Intercept X, Malwarebytes Anti Exploit, Palo Alto Networks Traps

Squiblydoo AppLocker Bypass

Similar to other whitelist defeating attacks, Squibydoo leverages operating system capabilities to run arbitrary scripts even on machines in full lockdown where only authorized scripts are intended to be run.

Vendors supporting this mitigation technique: Sophos Intercept X

Comparison

The following is an overview of exploit mitigations available in various security products, composed from datasheets, manuals, and product observations.

Memory Mitigations

	Sophos Intercept X	ESET Endpoint Security	Kaspersky Endpoint Security	McAfee Endpoint Security	Symantec Endpoint Protection	Trend Micro OfficeScan	Webroot Endpoint Protection	CylancePROTECT	Microsoft EMET	Malwarebytes Anti-Exploit	Palo Alto Networks Traps	CrowdStrike Falcon
Enforce Data Execution Prevention (DEP) Prevents abuse of buffer overflows	■			■					■	■	■	■
Mandatory Address Space Layout Randomization (ASLR) Prevents predictable code locations	■								■[1]		■	■[1]
Bottom Up ASLR Improved code location randomization	■								■	■		
Null Page (Null Dereference Protection) Stops exploits that jump via page 0	■		■						■			
Heap Spray Allocation Pre-allocated common memory areas to block example attacks	■				■				■	■	■	■
Dynamic Heap Spray Stops attacks that spray suspicious sequences on the heap	■										■[2]	

Code Mitigations

	Sophos Intercept X	ESET Endpoint Security	Kaspersky Endpoint Security	McAfee Endpoint Security	Symantec Endpoint Protection	Trend Micro OfficeScan	Webroot Endpoint Protection	CylancePROTECT	Microsoft EMET	Malwarebytes Anti-Exploit	Palo Alto Networks Traps	CrowdStrike Falcon
Stack Pivot Stops abuse of the stack pointer	■			■		■		■	■	■	■	
Stack Exec (MemProt) Stops attacker' code on the stack	■					■		■	■			
Stack-based ROP Mitigations (Caller) Stops standard Return-Oriented Programming attacks	■		■ 3	■ 1		■			■	■	■	
Branch-based ROP Mitigations (Hardware Augmented) Stops advanced Return-Oriented Programming attacks	■											
Structured Exception Handler Overwrite Protection (SEHOP) Stops abuse of the exception handler	■				■ 4				■ 2			
Import Address Table Filtering (IAF) (Hardware Augmented) Stops attackers that lookup API addresses in the IAT	■								EAF EAF+			
Load Library Prevents loading of libraries from UNC paths	■								■	■	■	
Reflective DLL Injection Prevents loading of a library from memory into a host process	■										■	
VBScript God Mode Prevents abuse of VBScript in IE to execute malicious code	■								■	■	■ 5	
WoW64 Stops attacks that address 64-bit function from WoW64 process	■											
Syscall Stops attackers that attempt to bypass security hooks	■											
Hollow Process Stops attacks that use legitimate processes to hide hostile code	■										■	
DLL Hijacking Gives priority to system libraries for downloaded applications	■											

Exploits Explained: Comprehensive Exploit Prevention and Vendor Offerings

	Sophos Intercept X	ESET Endpoint Security	Kaspersky Endpoint Security	McAfee Endpoint Security	Symantec Endpoint Protection	Trend Micro OfficeScan	Webroot Endpoint Protection	CylancePROTECT	Microsoft EMET	Malwarebytes Anti-Exploit	Palo Alto Networks Traps	CrowdStrike Falcon
Application Lockdown Stops logic-flaw attacks that bypass mitigations	■									■	■ 1	
Java Lockdown Prevents attacks that abuse Java to launch Windows executables	■									■	■	
Squiblydoo AppLocker Bypass Prevents regsvr32 from running remote scripts and code	■				■							
CVE-2013-5331 & CVE-2014-4113 via Metasploit In-memory payloads: Meterpreter & Mimikatz	■	■	■						■		■	

1 Based on ASLR functionality offered by Windows, available only in Windows Vista and newer versions of Windows

2 32-bit NOP sled and Polymorphic NOP sled only; no Flash Vector heap spray detection and not on 64-bit versions of Windows

3 32-bit ROP mitigation on WinExec() function only, not on 64-bit versions of Windows

4 Based on SEHOP functionality offered by Windows, available only in Windows Vista Service Pack 1 and newer versions of Windows

5 Human defined restrictions based on folders & child processes; high-maintenance, not behavior-based

United Kingdom and Worldwide Sales
Tel: +44 (0)8447 671131
Email: sales@sophos.com

North American Sales
Toll Free: 1-866-866-2802
Email: nasales@sophos.com

Australia and New Zealand Sales
Tel: +61 2 9409 9100
Email: sales@sophos.com.au

Asia Sales
Tel: +65 62244168
Email: salesasia@sophos.com

Oxford, UK

2017-12-1 WP-NA (MP)