SANS Institute
InfoSec Reading Room

Server Security: A Reality Check

Sponsored by Bit9

Server Security: A Reality Check

March 2014

A SANS Analyst Whitepaper

Written by Jake Williams

Case Study of an Advanced Attack ***PAGE 2***

Recommended Server Security Controls ***PAGE 5***

Why Servers Are Different ***PAGE 8***

Stepping Through the Compromise ***PAGE 9***

Review of ROI on Proactive Protection ***PAGE 10***

Introduction

Server security is paramount to any organization. All too often, network defenses focus entirely on the perimeter, leaving networks looking like pieces of candy—having a hard outer shell and a soft gooey inside. In today's threat landscape, defenders must operate under the presumption of compromise. Attacks involving spearphishing, social engineering, weak bring your own device (BYOD) security, poor physical security of end user equipment (for example, traveling laptops) and user carelessness contribute to the initial compromise of user endpoints. Once user endpoints are compromised, attackers typically use these as stepping off points to compromise other network assets.

The ultimate goal of most attacks on corporate infrastructure is theft of intellectual property (IP) and confidential customer data, such as personally identifiable information (PII), payment card information (PCI) and personal health information (PHI). Throughout this paper, the term *IP* will be used to represent all classes of sensitive data that organizations are required to keep confidential. However, IP is not normally stored on end user machines (and even when it is, it isn't concentrated there). Attackers know that IP is stored on the organization's servers. These servers are typically well protected from external threats, but once an attacker compromises a single endpoint, perimeter defenses are rendered moot. The attack can now proceed from the vantage point of an insider. Because of the sheer number of user endpoints, with various configurations (and vulnerabilities), a wise network defender *must* architect server defenses assuming that one or more of the endpoints in the environment has been already been compromised.

Case Study of an Advanced Attack

This case study walks through the steps attackers use to compromise a typical corporate network, locate critical IP, maintain covert access and exfiltrate IP from the network. If you recognize this attack (and it feels a little too close to home), you are not alone. Parts of this same scenario play out in network after network, day after day. Advanced attackers are comfortable with their playbook and, unless defenders deploy new defenses, attackers have no reason to change.

The initial source of compromise could be anything (spearphishing, watering hole attack, physical intrusion, insecure wireless access and so on). The odds that a compromise of one or more endpoints has not already occurred (or is unlikely to occur) are not in the defender's favor. Attackers know that gaining access to an endpoint is simply a waiting game. Even if your defenses are able to block the first hundred (or ten thousand) attacks, it takes only one successful attack to gain a foothold. Given that 95 percent of state-sponsored attacks establish initial network access via spearphishing,[1] such an attack is a logical place for our case study to start. Figure 1 provides a diagram of a typical network and the point of entry for an attack, this one via spearphishing.

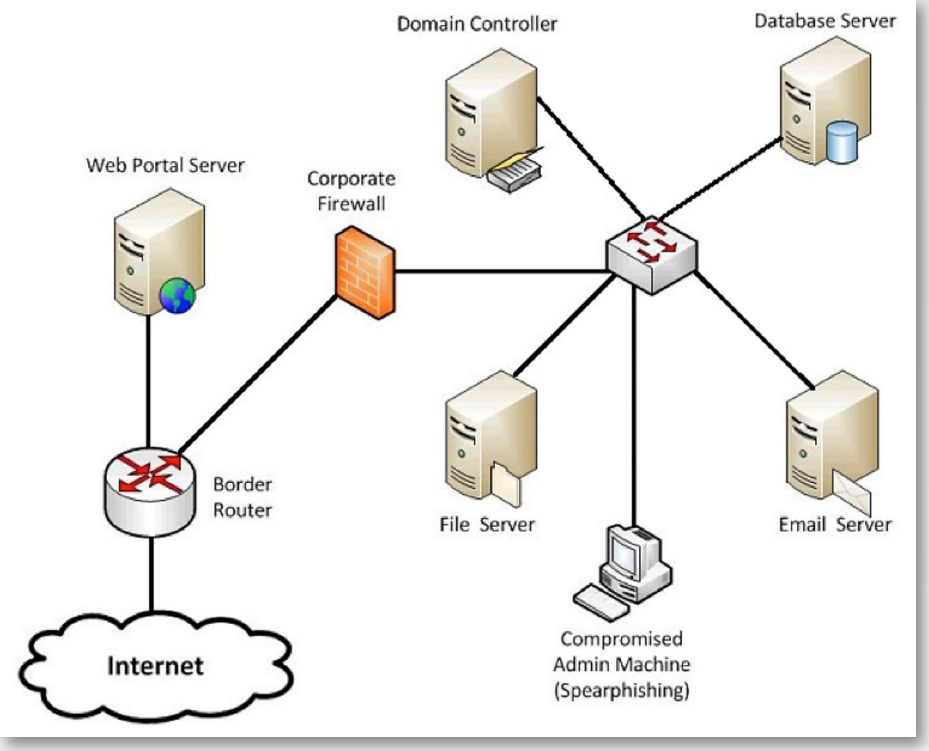

Figure 1. Sample Network with Compromised Systems

After attackers gain a foothold in the network, the fun begins. Attackers will attempt to leverage endpoint access to compromise servers that are not directly accessible from the Internet. This is when server security makes the difference between losing all of a company's IP or detecting the attack as it escalates. All too often, defenders think that server security impacts only whether or not the server is compromised (and the IP lost). However, properly secured (and instrumented) servers can be used not only to repel an attack, but also to detect (and remediate) the compromised endpoint.

[1] www.verizonenterprise.com/resources/reports/rp_data-breach-investigations-report-2013_en_xg.pdf

Case Study of an Advanced Attack (CONTINUED)

In our scenario the attackers leverage the access from a compromised administrator machine to gain access to the domain controller. Once there, the attackers download software to dump password hashes of all users for offline cracking. Upon inspecting the server configurations, the attackers realize that the domain requires complex passwords, making offline cracking difficult. They then decide to install software on the domain controller that will capture all plaintext passwords as they are changed. They do this by downloading a special password filtering DLL (which loads into `lsass.exe`) and modifying a registry key to load the malware. Now, whenever any user changes his or her password, the attackers get a plaintext copy immediately—no cracking required. Finally, the attackers load a backdoor that listens for connections and gives the attacker a shell when requested.

Armed with credentials and domain-level access, the attackers now turn to stealing IP and exploit the file server. In our example they are specifically looking for engineering data. The attackers install a new backdoor on the file server, so every time files with Visio extensions are updated, they are automatically compressed, encrypted and exfiltrated from the network.

Attackers may also look for source code stored on the file server. This code may contain encryption keys or other sensitive data that allows the attackers to gain additional access. They configure the new backdoor to load on every boot using an `autorun` registry key. Rather than being an asset, the corporate file server has now become a liability—a one-stop shop for locating and exfiltrating critical IP.

Unfortunately for the attackers, all of the IP isn't on the file server. Much of the critical IP they seek is stored on the database server. Some sensitive data may be stored using column- or table-level encryption. However, armed with the keys recovered from application code, the encryption is no problem. Once on the database server, the attackers pilfer data using built-in commands.[2] They also install a backdoor and new database modules that exfiltrate data as it is updated.

This practice is a trend seen increasingly among attacks: Attackers compromise the server, survey for critical IP and then install custom tools to exfiltrate data of interest automatically. This way, as the target creates new IP, the IP is automatically sent to the attacker at a waiting hop point. These tools are increasingly modular and custom assembled for the target. Because of this, they tend to have zero detection rates with traditional antivirus software, which focuses on blocking known bad software only.

Not satisfied that they have completely owned the network, attackers pivot from the domain controller to the email server. Here, they upload custom tools to search the server for mail containing specific keywords dealing with IP. Next, they change the mail server configurations so they receive carbon copies of the security administrator's email on a free webmail account. This provides the attackers with an early warning system in case they are detected. If attackers learn they have been detected, they are likely to modify their techniques and deploy different tools, forcing incident response teams to restart detection operations from ground zero.

[2] The Verizon Data Breach Investigations Report indicates that 63 percent of malware installed during breaches can exfiltrate data.

Finally, the attackers pivot to the partner network web portal server. This organization wasn't the end target after all! The attackers change web pages on the server to return malicious Java applets, compromising the organization's industry partners. They use the private keys recovered earlier from the file server to sign the malicious applets. Because the partner network already trusts the organization's code-signing certificate, these new malicious applets are also implicitly trusted. The attack goes off without a hitch! The attackers have successfully used a partner organization (often a contractor) to pivot into another, higher-security network. If this sounds far-fetched, consider that a very similar attack was used to compromise Twitter, Facebook, and Apple.[3] A site with developer libraries was compromised, and it distributed malicious code to these partners.

Table 2 summarizes the complete attack.

Table 2. Server Type Versus Attacker Motivation/Data Loss

Server Type	Attacker Motivation/Data Loss
Domain controller	Account password hashes, password filter installation, group policy modification
File server	IP, regulated data
Database server	Confidential records, column-level encryption
Email server	Corporate communications, business data, early warning of detection
Web server	Infect partner organizations

[3] https://threatpost.com/why-watering-hole-attacks-work-032013/77647

Recommended Server Security Controls

Let's look at some of the most effective server security controls. Brief definitions appear in the sidebar.

Log Monitoring

The first, and most important, security control for servers is continuous log monitoring. This, of course, is nothing new. Continuous monitoring won't help prevent a compromise, but it will help detect one in progress and allow responders to mitigate the threat. Of course, monitoring matters only if the event of interest is logged in the first place. If the logged event is easy to find because it is in a smaller pool of events, all the better.

Too often, systems and security administrators equate log review with the painful process of wading through gigabytes of irrelevant data to find the one piece of signal in all of the noise. But it doesn't have to be that way. By ensuring that those log entries that are likely to be relevant filter to the top, detection of compromises becomes phenomenally easier. This isn't to say that logging should be restricted to very small numbers of events. Rather, set your log management system to emphasize events that either have historically contributed to compromise detection or are most likely to flag specific activity related to compromise.

Server Security Controls Defined

This sidebar introduces some basic server security controls. How many of these are in use in your organization today?

Log monitoring—Server admins monitor logs for anomalous events that may indicate compromise.

Application control—Monitor application execution and log or block applications not explicitly authorized for execution (sometimes referred to as *whitelisting*).

Advanced application control—Monitor applications at a more granular level by validating expected command-line arguments. Allow application control to be used effectively with scripting languages, such as PowerShell.

File integrity monitoring—Monitor checksums of key files that should not regularly change. Alert when changes to key files are detected.

Registry rules—Monitor key registry keys and values for unauthorized changes. Alert when changes are discovered.

Application Control

Application control is another recommended server security control. Application control can be used to prevent unauthorized applications from executing on a server and log attempts made for unauthorized execution. Attackers often upload tools to newly compromised machines to gather credentials, harvest IP and establish persistence. Given that servers have a well-defined set of normal applications, those running outside of this set are immediately suspect. Because end users normally do not have the permissions necessary to execute applications on servers, new application executions can be attributed to either a small set of server administrators or an attacker. In either case, the root cause can be determined quickly. In the case of an attacker, this detection will normally lead back to the end user workstation that was originally compromised.

The difference in outcomes couldn't be clearer. With no application control, the attacker pivots from the end user workstation to the server undetected. With effective application controls in place, the pivot is unsuccessful *and* the compromise of the end user workstation (where the pivot operation began) is detected.

Advanced Application Control

Traditional application control is only granular enough to allow or block execution of a particular application. However, some applications can be used to achieve either authorized or unauthorized actions, depending on the command-line parameters used. The most prolific example of this type of application are scripting language interpreters.

Scripting languages, particularly PowerShell, add a new dimension to attacks on servers. Entire attacker tool frameworks have been built using various scripting languages. Because PowerShell is installed by default on modern Windows servers, it deserves the most attention. Although PowerShell configurations can restrict execution to only signed scripts, this feature can be disabled at the command prompt. Even Microsoft admits that this feature is not intended to prevent all illicit use of PowerShell; rather, it is designed to prevent users from being tricked into running unsigned scripts inadvertently.

Unfortunately, traditional whitelisting implementations of application control cannot be used to block execution of PowerShell scripts. Administrators rely on PowerShell too heavily to allow its use to be blocked entirely. Application control, then, must be granular enough to differentiate between legitimate administrator scripts and illegitimate attacker scripts.

If application control has this level of granularity, it can even prevent attacks in which the entire script is placed on the command line. Attackers often employ this technique to avoid signing restrictions and, because scripts written to a file are subject to inspection, scanning by antivirus software. Therefore, we need application control that recognizes not only the application being run, but also the command-line arguments passed to the application.

File Integrity Monitoring

Once attackers compromise key systems, they want to make sure they have a way back in. In some cases, this is accomplished not by installing new software, but by making changes to the configuration files of existing software.

An excellent example of this on the Windows platform is when an attacker modifies an existing web application on a compromised web server to include an ASP shell. Such a web application allows the attacker to run arbitrary commands on the server using a web browser. Because the ASP shell runs inside the existing Internet Information Services (IIS) web server process (rather than actually executing as its own process), application control would not immediately catch the modified web application. However, the attacker would want the web application to have increased permissions and would modify the website configuration file. Because this file is protected with file integrity monitoring, the system can detect the attacker immediately. Not only is the intrusion detected, but because file integrity monitoring was in place, incident responders are also able to quickly determine how many other files were modified.

Registry Rules

Registry rules are another critical server control. As powerful as file integrity monitoring can be, it can't help prevent changes to registry hives (because the registry changes constantly under day-to-day use). Unfortunately, many applications store their configurations exclusively in the registry, rendering file integrity monitoring useless to protect these application configurations. However, attackers may change these registry configuration settings to achieve persistence on a compromised machine.

By enabling a registry monitoring application, defenders gain the benefits of file integrity monitoring, but at a much more granular level than was previously possible. Certainly some security suites already monitor (and/or prevent) changes to well-known `autorun` locations. However, new `autorun` locations are discovered regularly by malware authors.

The most effective approach clearly is to take inventory of the registry and then prevent changes to those areas that are not prone to change. Unlike the blacklisting approach used by most security software, registry integrity monitoring at the key and value levels protects against threats that are as yet unknown. Furthermore, this level of monitoring may enable the security team to determine what specifically was changed. This may aid the team in making the decision to resume operations without restoring to a known clean backup.

Why Servers Are Different

Among seasoned information security professionals, application control is a touchy subject, to say the least. Many have experienced problems implementing it (or know of those who have). Stories abound of having to configure (and manage) ever-growing lists of exception rules to deal with business uses.

But the problems with the deployment of application control are largely unique to end user machines. Such devices are in a near-constant state of flux as new business requirements are levied. Individual computers also tend to have many more third-party applications (file viewers, productivity applications and so on) installed than an average server. Each of these applications requires security and functionality updates, and with each update, the definitions and profiles for application control need to be updated.

Consider for a moment, however, how infrequently we make significant application changes on servers. In most environments the traditional objections to application control don't make sense in the datacenter. As a rule, changes to servers are fewer and farther between than changes made to end user machines.

Further, the list of software applications that must be authorized to execute on the server is normally much smaller per server than per workstation. Do you need to support three different web browsers (and associated plug-ins) on the server to support different web applications? Of course not. The reality is that servers are an ideal place to roll out application control: They have the smallest number of applications to support; they change the least; and maintenance windows are well scheduled when changes are required.

Even if you determine that application control can't work for all of your systems, shouldn't it protect the most critical systems on the network? Of course it should. To argue otherwise is like saying that bulletproof vests are useless to law enforcement because they don't protect the entire body. Application control in the datacenter, similar to bulletproof vests, protects the most critical portions of the business from harm.

Stepping Through the Compromise

Having examined the case study attack and potential server controls, let's consider the same attack if the target organization had application control enabled on its servers.

Assuming that application control is deployed only on servers, it won't detect the initial spearphishing compromise on the end user machine. However, application control will prevent lateral movement from the compromised server to the domain controller. It will also prevent the execution of the password dumping tool and the loading of the unauthorized password filtering DLL. Best of all, because those actions are blocked, it gives security personnel a chance to respond to alerts, and the initial compromise activity is easily tracked back to the compromised machine. Even if DLL monitoring isn't configured as part of application control (it should be), the loading of the password filtering DLL is performed via a registry change. Because this registry key should not regularly change, it is prevented by registry rules. When the attackers attempt to change the registry key, they are blocked from doing so, and the attempt is logged.

The same story plays out at the file server. Remember that backdoor that automatically exfiltrated new engineering data as it was placed on the file server? It is also mitigated by application control. The attackers are unable to install their tools needed to gain full system privileges, and as result, it is far more difficult for them to compromise the IP. Attackers could still use compromised workstations to scan exposed file shares without running malware on the file server itself. However, in most configurations, this creates voluminous logs. Further, attackers are limited to the permissions of the user whose workstation they have compromised. The takeaway is that when attackers can't run their malware on the targeted servers, they have to work markedly harder to steal your IP.

Although application control prevents attackers from executing malware on the database server itself, they can use remote connections to the database server's listening ports to pilfer data. However, similar to their approach to the file server, the attackers prefer direct server access so they can bypass any restrictive account controls. Changes to database configuration files are protected by file integrity monitoring. Application control and registry rules also prevent custom malware from being installed on the database server to exfiltrate IP or the installation of a backdoor.

A similar story evolves at the mail server. Application control prevents the attackers from loading backdoors and mail-filtering software on the server, and file integrity monitoring prevents the attackers from changing email server configurations. Securing the company's internal mail server is paramount, and its importance cannot be overstated. Advanced attackers may compromise the mail server and monitor administrators' mailboxes for any signs that their activity has been detected. Further, attackers may use the organization's mail servers to send spam or spearphishing emails, compromising other networks. Using application control and file integrity monitoring saves the day.

The web server, however, is the key to this attack. Everything up to this point has primarily been a means to an end. Recall that the attackers seek to distribute malicious signed Java applets to business partners. However, because the attackers were unable to compromise the file server, they were unable to locate the highly secured code-signing certificates. Further, they are unable to make changes to the website due to file integrity monitoring. Finally, any application configuration changes stored in the registry are also blocked by registry rules.

Review of ROI on Proactive Protection

In today's environment of fiscal conservatism, justifying expenditures on new security measures is particularly challenging. This is at least in part because it is difficult to establish an accurate return on investment (ROI) on technology that is purely preventive. Organizations don't see any immediate increased productivity or other benefits on the expenditure: The return is realized only in the form of compromises prevented (or mitigated).

Cost of Breach

One good metric for assessing ROI on security spending is to consider what companies do after a breach. The Ponemon Institute surveyed organizations that had been breached and found that 60 percent increased their IT security budget and 28 percent added IT security staff in response.[4] The question fundamentally becomes an issue of whether organizations will pay now or pay later.

Another study found that organizations with a strong security posture could expect to reduce their overall per-record data breach costs by up to $34 per record (from a baseline average of $199 per record).[5] Adding security controls, such as application control, file integrity monitoring and registry rules, to server assets (where they matter most) definitely increases a company's security posture.

Even in cases where companies feel they have no IP to lose, loss of reputation is still an issue. Each industry has its own costs for attracting new customers, and they vary too widely to effectively establish an overall norm. However, it should be noted that 15 percent of customers notified of a loss of their private data say they will discontinue their relationship with the vendor who lost it (and 39 percent say they will consider discontinuing it).[6] Losing 15 percent or more of your customer base overnight due to a single compromise is a huge short-term issue.

The Ponemon Institute also attempted to quantify the actual dollar cost of brand reputation losses. They found that after a breach, the value of a brand decreased by between $184 and $330 million.[7] With an average brand value of $1.5 billion, this represents a minimum loss of 12 percent of the brand value (increasing to 22 percent in some cases). Organizations wishing to analyze the ROI of proactive monitoring may need to look no further than their own brand's value. The costs associated with gaining customers to replace lost ones can be significant in the long term. One such retailer, Target, offered 10 percent discounts on retail purchases following a breach in an attempt to attract and retain customers.[8] Target also suffered a 3–4 percent drop in sales in an important holiday shopping week. This was obviously a huge blow in an already down economy.

Organizations that experience a breach can expect to experience considerable productivity losses. Productivity losses were the number one reported negative experience resulting from a breach.[9] This is perhaps not surprising because, after a breach, security managers pull staff members from their normal duties to investigate what went wrong and implement corrective actions.

---

[4] www.ponemon.org/local/upload/file/Aftermath_of_a_Data_Breach_WP_Final%20.pdf

[5] www4.symantec.com/mktginfo/whitepaper/053013_GL_NA_WP_Ponemon-2013-Cost-of-a-Data-Breach-Report_daiNA_cta72382.pdf

[6] www.experian.com/assets/data-breach/brochures/ponemon-notification-study-2012.pdf

[7] http://press.experian.com/United-States/Press-Release/new-survey-by-the-ponemon-institute-finds-that-data-breaches-can-cause-lasting-and-costly-damage.aspx?&p=1

[8] www.businessweek.com/news/2013-12-24/target-seen-losing-customer-loyalty-in-wake-of-data-breach-1

[9] www.ponemon.org/local/upload/file/Aftermath_of_a_Data_Breach_WP_Final%20.pdf

So the costs of a breach are significant. But what about the costs of proactive monitoring to prevent a breach from happening in the first place? This cost, of course, depends on the quantity of assets (computers) that must be protected and monitored.

Cost of Proactive Monitoring

We assume a yearly baseline cost of $100 U.S. per software agent for each node that must be protected. We further assume a monitoring cost of $50 U.S. per year for each protected node. This cost includes man-hours devoted to monitoring and reviewing alerts. Finally, we assume that new monitoring hardware is also required. We assign this new hardware a cost of $5,000 for all monitored nodes. It is important to note that per-node software licensing, software deployment and licensing costs will decrease as the size of the installation increases.

Obviously, many factors influence the costs to deploy monitoring in a given organization. Also, the costs to initially deploy this type of solution are generally higher than costs incurred in subsequent years. Table 3 provides a look at the costs of proactive monitoring in the initial year.

Table 3. Cost of Proactive Monitoring

Number of Nodes	Software	Monitoring	Hardware	Total
20	$2,000	$1,000	$5,000	$8,000
50	$5,000	$2,500	$5,000	$12,500
100	$10,000	$5,000	$5,000	$20,000
200	$20,000	$10,000	$5,000	$35,000
500	$50,000	$25,000	$5,000	$80,000
1,000	$100,000	$50,000	$5,000	$155,000

Table 4 looks at the costs of some of the more well-publicized breaches in the past few years.

Table 4. Cost of Breaches

Breach	Records Lost	Cost
Heartland Payments [10]	130 million	$12.6 million
Global Payments[11]	1.5 million	$92.7 million
Target[12]	40 million	$1 billion (estimated)

[10] www.bankinfosecurity.com/card-fraud-case-sheds-light-on-breaches-a-5946

[11] www.bankinfosecurity.com/card-fraud-case-sheds-light-on-breaches-a-5946

[12] www.twincities.com/business/ci_25029900/analyst-sees-target-data-breach-costs-topping-1

The Bottom Line

The costs for a proactive monitoring program are a definite expenditure, whereas the costs of a breach are uncertain. Considering that the average cost of a breach is $5.4 million, proactive monitoring to prevent a breach that costs a small fraction of that may be easily justified.[13] Organizations should carefully evaluate their overall risk tolerance. Many organizations rely heavily on the confidentiality of their IP and customer records to remain in business. For these organizations, the cost of protecting this IP from disclosure to outside parties (attackers) is perhaps easier to absorb. For organizations that face steep regulatory costs if they lose confidential data (for example, due to HIPAA or other regulatory statutes), again the cost/benefit analysis should be clear.

As described earlier in the paper, organizations can stop many attacks in their tracks simply by increasing security on servers. Obviously, organizations get the best security posture by enhancing security across the entire organization. But they can get the biggest bang for their buck by increasing security where it matters most—at the server level. This also reduces the number of nodes that must be protected (lowering the overall cost). Assuming the costs in Table 3 for a large server deployment (500 nodes), the first-year costs are only $80,000. Considering that the average breach costs $5.4 million, this first-year cost to protect an organization with 500 servers represents only 1.4 percent of that cost. Many organizations have far fewer than 500 servers, meaning the costs could be substantially lower. Integrated as part of a total security package, the costs for increased server security are very easily justified.

[13] www4.symantec.com/mktginfo/whitepaper/053013_GL_NA_WP_Ponemon-2013-Cost-of-a-Data-Breach-Report_daiNA_cta72382.pdf

Conclusion

Security breaches happen. Eventually, every organization will experience a compromise. The only question is how crippling the breach will be. Will the attackers successfully compromise sensitive IP? Will they exfiltrate regulated data? Or will the attack be detected quickly in the early infiltration stages? Clearly, detecting an attack early and preventing attackers from moving laterally to servers is the key to minimizing the damage from a breach.

In this paper we walked through a potential compromise scenario. This scenario is typical of real-world compromises happening today. Then, we examined server security controls and how they can be used to prevent these attacks and quickly detect compromised machines. Finally, we made the case for the ROI on server security controls.

Adding server security controls can have significant financial implications in preventing compromises. At the very least, these controls can minimize the IP lost when a compromise does occur. Organizations that examine their current server security controls will likely determine that the addition of application control and file integrity monitoring makes sense for their environments.

About the Author

Jake Williams is the chief scientist at CSRgroup computer security consultants and has more than a decade of experience in secure network design, penetration testing, incident response, forensics and malware reverse engineering. Before joining CSRgroup, he worked with various government agencies in information security roles. Jake is a two-time victor at the annual DC3 Digital Forensics Challenge and a SANS course author and certified instructor.

SANS would like to thank this paper's sponsor:

Last Updated: July 19th, 2018

Upcoming SANS Training

Click here to view a list of all SANS Courses

SANS Riyadh July 2018	Riyadh, SA	Jul 28, 2018 - Aug 02, 2018	Live Event
SANS Pittsburgh 2018	Pittsburgh, PAUS	Jul 30, 2018 - Aug 04, 2018	Live Event
Security Operations Summit & Training 2018	New Orleans, LAUS	Jul 30, 2018 - Aug 06, 2018	Live Event
SANS Hyderabad 2018	Hyderabad, IN	Aug 06, 2018 - Aug 11, 2018	Live Event
Security Awareness Summit & Training 2018	Charleston, SCUS	Aug 06, 2018 - Aug 15, 2018	Live Event
SANS Boston Summer 2018	Boston, MAUS	Aug 06, 2018 - Aug 11, 2018	Live Event
SANS San Antonio 2018	San Antonio, TXUS	Aug 06, 2018 - Aug 11, 2018	Live Event
SANS August Sydney 2018	Sydney, AU	Aug 06, 2018 - Aug 25, 2018	Live Event
SANS New York City Summer 2018	New York City, NYUS	Aug 13, 2018 - Aug 18, 2018	Live Event
SANS Northern Virginia- Alexandria 2018	Alexandria, VAUS	Aug 13, 2018 - Aug 18, 2018	Live Event
SANS Krakow 2018	Krakow, PL	Aug 20, 2018 - Aug 25, 2018	Live Event
Data Breach Summit & Training 2018	New York City, NYUS	Aug 20, 2018 - Aug 27, 2018	Live Event
SANS Chicago 2018	Chicago, ILUS	Aug 20, 2018 - Aug 25, 2018	Live Event
SANS Prague 2018	Prague, CZ	Aug 20, 2018 - Aug 25, 2018	Live Event
SANS Virginia Beach 2018	Virginia Beach, VAUS	Aug 20, 2018 - Aug 31, 2018	Live Event
SANS San Francisco Summer 2018	San Francisco, CAUS	Aug 26, 2018 - Aug 31, 2018	Live Event
SANS Copenhagen August 2018	Copenhagen, DK	Aug 27, 2018 - Sep 01, 2018	Live Event
SANS SEC504 @ Bangalore 2018	Bangalore, IN	Aug 27, 2018 - Sep 01, 2018	Live Event
SANS Wellington 2018	Wellington, NZ	Sep 03, 2018 - Sep 08, 2018	Live Event
SANS Amsterdam September 2018	Amsterdam, NL	Sep 03, 2018 - Sep 08, 2018	Live Event
SANS Tokyo Autumn 2018	Tokyo, JP	Sep 03, 2018 - Sep 15, 2018	Live Event
SANS Tampa-Clearwater 2018	Tampa, FLUS	Sep 04, 2018 - Sep 09, 2018	Live Event
SANS MGT516 Beta One 2018	Arlington, VAUS	Sep 04, 2018 - Sep 08, 2018	Live Event
Threat Hunting & Incident Response Summit & Training 2018	New Orleans, LAUS	Sep 06, 2018 - Sep 13, 2018	Live Event
SANS Baltimore Fall 2018	Baltimore, MDUS	Sep 08, 2018 - Sep 15, 2018	Live Event
SANS Alaska Summit & Training 2018	Anchorage, AKUS	Sep 10, 2018 - Sep 15, 2018	Live Event
SANS Munich September 2018	Munich, DE	Sep 16, 2018 - Sep 22, 2018	Live Event
SANS London September 2018	London, GB	Sep 17, 2018 - Sep 22, 2018	Live Event
SANS Network Security 2018	Las Vegas, NVUS	Sep 23, 2018 - Sep 30, 2018	Live Event
SANS DFIR Prague Summit & Training 2018	Prague, CZ	Oct 01, 2018 - Oct 07, 2018	Live Event
Oil & Gas Cybersecurity Summit & Training 2018	Houston, TXUS	Oct 01, 2018 - Oct 06, 2018	Live Event
SANS Brussels October 2018	Brussels, BE	Oct 08, 2018 - Oct 13, 2018	Live Event
SANS Pen Test Berlin 2018	OnlineDE	Jul 23, 2018 - Jul 28, 2018	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced