HACKER SECRETS REVEALED

FIVE LESSONS LEARNED FROM SECURITY ASSESSMENTS

The technical objective of security assessments is to emulate an outside adversary to get access into an internal network, escalate privileges and obtain sensitive information. The intent is not to find every single vulnerability in the way that a vulnerability scan might do, but rather to find some of the vulnerabilities that exist, and attempt to exploit those.

MOTOROLA SOLUTIONS

INTRODUCTION

Every year, Motorola Solutions conducts hundreds of cybersecurity assessments, including penetration testing, for a wide range of commercial and public sector clients. Many of these organizations share similar weaknesses in their people, processes and technology. But each assessment also presents new technical challenges for us to solve.

Our research has identified the attack vectors bad actors most commonly use to get initial access to a network and then infiltrate the rest of the organization. We offer actionable recommendations on how to best combat each scenario to help defenders better understand attacker patterns and improve their cybersecurity posture.

In this white paper, we discuss our findings from external pen tests, also known as ethical hacking, against enterprise clients who have already implemented standard security best practices such as two-factor authentication (smart cards), identity access management controls, restricted administrative privileges and spam filtering.

As we share our top five technical findings and lessons learned from external assessments, we'll reveal weaknesses our testers exploited and offer vendor-neutral solutions for resolving each of these issues. The topics we'll discuss include phishing, Kerberoasting, administrative passwords on file shares and misconfigured local administrative privileges.

We'll also explain why insufficient network segmentation on its own isn't an exploitable vulnerability, but how lack of network segmentation can open your organization's internal attack surface.

LESSON #1 - HOW INSUFFICIENT NETWORK SEGMENTATION INCREASES YOUR SECURITY RISK

Every year, Motorola Solutions conducts hundreds of cybersecurity assessments, including penetration testing, for a wide range of commercial and public sector clients. Many of these organizations share similar weaknesses in their people processes, and technology. But each assessment also presents new technical challenges for us to solve.

Our research has identified the attack vectors bad actors most commonly use to get initial access to a network and then infiltrate the rest of the organization. We offer actionable recommendations on how to best combat each scenario to help defenders better understand attacker patterns and improve their cybersecurity posture.

In this white paper, we discuss our findings from external pen tests, also known as ethical hacking, against enterprise clients who have already implemented standard security best practices such as two-factor authentication (smart cards), identity access management controls, restricted administrative privileges and spam filtering.

As we share our top five technical findings and lessons learned from external assessments, we'll reveal weaknesses our testers exploited and offer vendor-neutral solutions for resolving each of these issues. The topics we'll discuss include phishing, Kerberoasting, administrative passwords on file shares and misconfigured local administrative privileges.

We'll also explain why insufficient network segmentation on its own isn't an exploitable vulnerability, but how lack of network segmentation can open your organization's internal attack surface.

Proper network segmentation can be very difficult to implement correctly, especially on networks that have existed without it for many years.

ABUSE DURING PRIVILEGE ESCALATION PHASE

The escalation phase takes place after we have gained initial access. One of the most common methods we use to get that access is through phishing emails to selected targets within in the organization. Once we gain initial access, we are usually impersonating a regular non-IT employee (someone without administrative rights). This type of user should only be able to access a few servers and file shares that they need to perform their day-to-day work functions. For example, an initial phishing victim might get us access to an HR user's workstation. A user in human resources typically would have no reason to attempt to access a web server within the IT department, or a payroll system within the finance department.

However, in organizations that do not implement any network segmentation, we are immediately able to see every server (file shares, SQL servers, web servers) and workstation that is connected to the internal network. This simplifies our job because we can start looking for misconfiguration anywhere in the organization to elevate our privileges and gain additional access. For example, we might initially use phishing to compromise the workstation of a user in the organization's Washington, D.C. marketing department. Then, due to lack of network segmentation, we may identify cleartext credentials on a share in Boston that we could use against a web application in Salt Lake City.

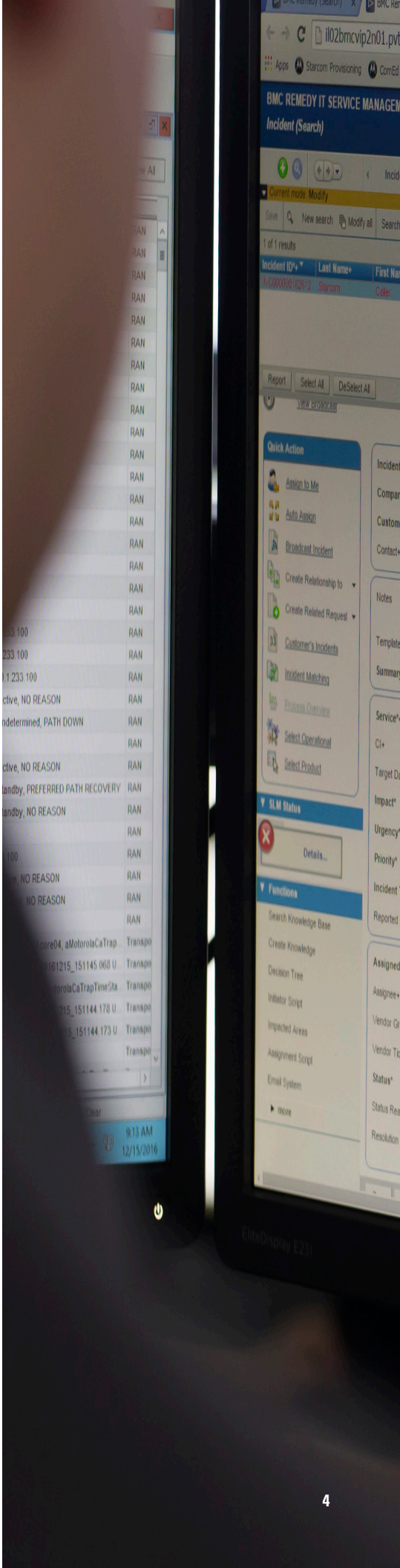
ABUSE DURING THE POST-EXPLOITATION PHASE

The post-exploitation phase starts after we have gained administrative access within the organization, meaning we have obtained credentials that allow us to log into any domain-joined host. The purpose of this phase is to "show impact" and demonstrate the risk to the organization that comes from the vulnerabilities and misconfigurations previously exploited during the assessment. We usually exfiltrate Personally Identifiable Information (PII) to show the leadership team how someone with unfettered access can get to valuable, confidential information once they're on the network.

When network segmentation hasn't been implemented, we can easily jump anywhere we need to go. For instance, we can move from a summer intern's workstation directly to the workstation of a system administrator in charge of a server holding social security numbers. Even though there is no business reason for the intern to access the sysadmin's workstation, without proper network segmentation, there is nothing stopping those hosts from communicating – and from the intern being able to pull another employee's information.

Proper network segmentation (allowing only required communication) can considerably reduce your internal attack surface. When handled correctly, network segmentation minimizes the number of hosts an attacker can potentially exploit, and inhibits an attacker's ability to spread laterally within an organization.

One best practice is to logically group systems based on work-task function, for example grouping HR people into one work group, and VLAN them apart from another group such as the sales work group. Moreover, it is critical that every work-task group should have their own file share. With that structure, segmentation can also help defenders detect malicious behavior within their network by alerting on hosts that are attempting to access systems they have no business trying to access.



LESSON #2 - IDENTIFYING LOCAL ADMIN MISCONFIGURATIONS FOR DOMAIN PRIVILEGE ESCALATION

This section covers an element that we frequently abuse during the privilege escalation phase of our penetration testing assessments, particularly those involving public sector clients. This phase occurs after our operators have gained a foothold and established persistence within a client's internal network. At this point of the assessment, our foothold into the network is in the context of a domain user (or more often, several domain users).

PRIVILEGE ESCALATION THROUGH LATERAL MOVEMENT

One of the tried and true methods of privilege escalation is to use your current access to spread laterally throughout the environment. This is done by taking advantage of misconfigurations at either the host level or through domain group misconfigurations. As pen testers, our objective is to repeatedly identify and move to a host that has a more privileged user logged in until we reach the desired level of access. This method usually requires several "hops" before we compromise a user of who meets our preferred level of privileges, typically Domain Administrators.

For example, we might initially gain access to a user in the Human Resources department through a phishing email. Then, we might attempt to leverage a misconfiguration of domain groups to compromise a user from a group of Workstation Administrators, who have admin rights over multiple workstations. Then we might use that access to compromise a workstation with a Domain Admin logged in. If our operators can gain administrative access to any host with a privileged user logged in, then we can use a tool like Mimikatz to dump the logged-on user's credentials and impersonate that user for the remainder of the assessment.

To spread laterally to another workstation or server in a domain, we generally need to have administrative access to the target host. Our preferred method of lateral movement is performed via a network login, which is transparent to any logged-in user. We use this network login to execute a command on a remote host that provides us full control of that target computer. The command is executed either by remotely creating and starting a custom service on the new host, or by using Windows Event Management (WMI) or Windows Remote Management (WinRM) to directly execute the command on the target host.

LOCAL ADMIN MISCONFIGURATIONS

In general, public sector organizations do a very good job of following standard security best practices, especially when it comes to restricting the administrative rights of non-privileged users. It's rare that we will assess a client that allows all users to run as a local Administrator. We also find that when organizations provide their users with "golden images" of their workstation (essentially prebuilt and pre-configured workstations), it drastically cuts down on us finding common Windows privilege escalation vulnerabilities, such as DLL hijacking, unrestricted service permissions, or enabled AlwaysInstallElevated policies.[1]

Keeping with standard security best practices, many public sector organizations have implemented Microsoft LAPS, which generates strong, unique passwords for the Built-In Administrator account (RID-500) on all hosts. This means we cannot simply pass-the-hash with this account to access other hosts. So, gaining administrative (SYSTEM level) access on a single host is far from the end of the privilege escalation phase. However, being able to spread to a couple additional boxes right off the bat can give us a big advantage in an assessment.

Even with all the effort put into restricting local administrator access, we still frequently find an "edge case" workstation or server that has been misconfigured (typically a smaller number of hosts) to allow local administrative privileges to an excessive number of users. This generally occurs due to an overly permissive Active Directory group being added to that host's local Administrators Group. Sometimes this happens due to an Active Directory group containing additional groups, which can quickly lead to a massive number of users that are now part of the group. Occasionally, we even find an old, forgotten host that has been configured to allow all "Domain Users" to be in the local Administrators group.

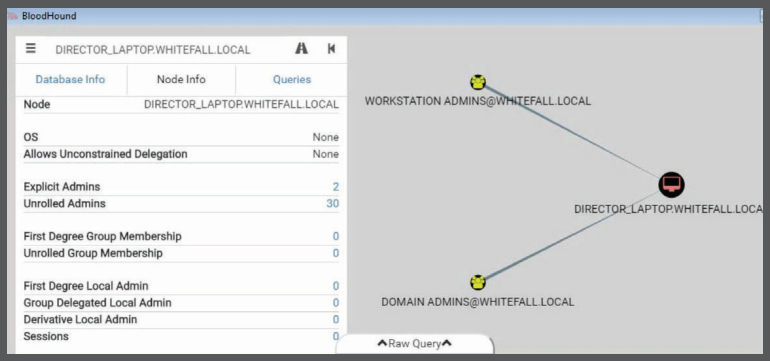


Figure 1: Admins with Director Laptop Access

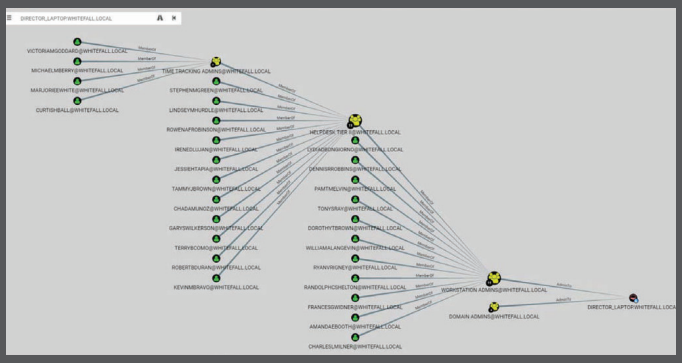


Figure 2: Access Through Unrolled Workstation Admin Groups

NESTED GROUP MEMBERSHIP MISCONFIGURATION EXAMPLE

To demonstrate how groups inside of groups can quickly become unruly, we've recreated a small Windows domain to highlight this misconfiguration. Let's look at who has local Administrative access over the Director's laptop in this organization, using an open-source tool called BloodHound. BloodHound is an open source tool originally created to assist network defenders and auditors to understand their systems more completely, but as we know now, tools that can help defenders can also help attackers.[2]

The screenshot below shows that the only Admins to the Director's Laptop are the groups "Domain Admins" and "Workstation Admins." At first glance, this looks fairly locked down; even the Director is not granted administrative access to their own laptop.

However, the "Unrolled Admins" attribute on the left side shows that there are actually 30 users who have administrative access over this host. When we unravel the "Workstation Admins" group membership, we see that it contains 11 explicit users and the "Helpdesk Tier II" group. Depending on the organization's policies, it may or may not be acceptable for all Tier II helpdesk employees to have administrative access to the Director's laptop. Looking further, we see that the "Helpdesk Tier II" group contains another 11 users, plus the "Time Tracking Admins" group. Now we are up to more than 20 users inside the "Workstation Admins" group, and have administrative access to the Director's laptop.

The "Time Tracking Admins" group is likely composed of Human Resource staff in charge of recording employee's working hours. Therefore, placing the "Time Tracking Admins" group inside of the "Helpdesk Tier II" group is almost certainly a misconfiguration.

In this instance, this misconfiguration represents a significant risk to the organization because an attacker only needs to compromise a "Time Tracking Admin" to get unrestricted administrative access to the Director's laptop. Effectively, any one of these 30 users could install and run malicious programs remotely on the Director's laptop, such as a keylogger, a screen capture tool, or a remote-control agent capable of exfiltrating sensitive data. This ultimately helps us focus down on our specific targets.

We find that when we assess an organization and spend a couple of weeks looking at the security controls, we know more about the network's security than the people who are working there fulltime. The misconfigurations that we take advantage of are usually the result of a problem that has been compounded by many small changes over time, and end up resulting in a major security hole.

Recommendations to keep in mind include carefully considering whether it is a necessity to add an Active Directory group to the Local Admin group; it should be a conscious decision that considers the security implications include management of the group. If it the decision is to add the group, be sure to document the actions thoroughly, including length of access required, specifics of the group added to which specific system and time and date.

Additionally, be sure to frequently query or audit your environment for overly permissive Active Directory groups. By routinely auditing and assessing the people, processes and technology present in a network, network defenders can stay abreast of misconfigurations such as those presented in this post, and increase their organization's overall level of security.

Our next finding focuses on the cyber risk involved with not securing clear text administrative passwords.

LESSON #3 - NAVIGATING CLEAR TEXT PASSWORD VULNERABILITIES

ACCESSING CLEAR TEXT ADMINISTRATIVE PASSWORDS

In the previous section, we showed how pen testers can use misconfigurations within Active Directory group management to escalate privileges. However, that technique is heavily dependent on having access to privileged or misconfigured accounts in the first place.

Next, we discuss another finding that we frequently take advantage of during both the privilege escalation and post-exploitation phases of our assessments. Having access to account credentials is a vital aspect of every penetration test, so we are always on the lookout for methods to obtain valid password credentials of key accounts within the organization. We use a variety of techniques to gain access, but one of the simplest and most reliable methods is to find clear text credentials on the internal network. Clear text passwords don't require decryption to be viewed.[3]

INSECURE CREDENTIALS STORAGE DURING PRIVILEGE ESCALATION

On some assessments, the user accounts we initially compromise from phishing emails enable us to spread laterally within the network and escalate the domain. In other assessments, our initial access does not grant administrative access anywhere in the organization. In some cases, organizations are adequately managing their Active Directory groups to avoid such misconfigurations, and in others, our phishing emails are unable to trick any of the organization's more privileged users. Regardless, there are some assessments where we can't immediately escalate the domain by abusing Active Directory groups. This is a strength for the organization. However, it makes our job as pen testers more difficult.

If we can't escalate the domain, the next step is to leverage our current access to get additional credentials within the organization. This is where share-hunting (finding internal file shares on the network) comes into play. Most organizations are doing a better job of locking down administrative access to internal hosts, but they may not always lock down access to internal file shares. Even though a normal user may only need access to a limited number of file shares for their day-to-day work, they may have been granted access to additional share drives.

We prefer to use the Invoke-ShareFinder function from the PowerShell script Powerview.ps1 to identify which file shares are available in the network.[4] This script identifies all hosts in the current domain, and queries each one directly to determine what shares are available. Furthermore, it can also attempt to read from each share to determine what shares the current user can access. The result is a comprehensive list of all shared drives that our compromised user can read. This list includes the primary file shares that employees access on a regular basis, plus old file shares that may be left over from legacy systems or misconfigured hosts that were not intended to share data.

Once we have a list of all shares, the next step is one of the most tedious parts of penetration testing—we spend time manually looking through shares for interesting files. There are some keywords we can trigger for automated searching (e.g., password, secret, config), but our testers frequently attain more valuable files by manually searching for them. Sometimes we will get lucky with a "passwords. txt" or "passwords.xlsx" file, but normally we end up finding forgotten domain administration scripts or system configuration files that contain hardcoded passwords. Once we collect credentials for various accounts, we will attempt to validate them and use them to spread laterally, while continuing to escalate our privileges within the domain.

One important thing to note is that most of our public sector clients have implemented multi-factor authentication for regular user domain accounts. This means that all users require a smart card and PIN to perform an interactive login. Although this does not stop us from using compromised accounts for lateral movement, it does cut down on the number of valid clear text passwords we find for domain users. However, many Service Accounts and Administrative Accounts do not user multi-factor and still use traditional username and passwords combinations for authentication. This makes those privileged accounts significantly easier for us to abuse—and for malicious hackers to exploit

INSECURE CREDENTIALS STORAGE DURING POST-EXPLOITATION

A critical part of any assessment is the post-exploitation phase. Not all executives fully comprehend the risks associated with an attacker gaining domain administrator privileges to their organization. Therefore, the purpose of this phase is to show impact to help executives better understand the risks that come from their organization's vulnerabilities and misconfigurations.

We find it most effective to show impact by identifying and exfiltrating highly protected information. Sometimes this takes the form of Personally Identifiable Information (PII) such as Social Security Numbers; other times it takes the form of sensitive budgeting and billing documents.

Most organizations take additional steps to safeguard their most sensitive information. Frequently, we find that systems containing sensitive information are not joined to the domain, but instead managed by separate, local accounts. Therefore, gaining domain administrator level access does not provide us direct access to sensitive data. In this case, we try to hunt down the administrators in charge of these systems to monitor how they access these systems (e.g., SSH, web applications) and the accounts they use.

Many Service Accounts and Administrative Accounts still use traditional username and passwords combinations for authentication, which makes those privileged accounts significantly easier for malicious hackers to exploit.

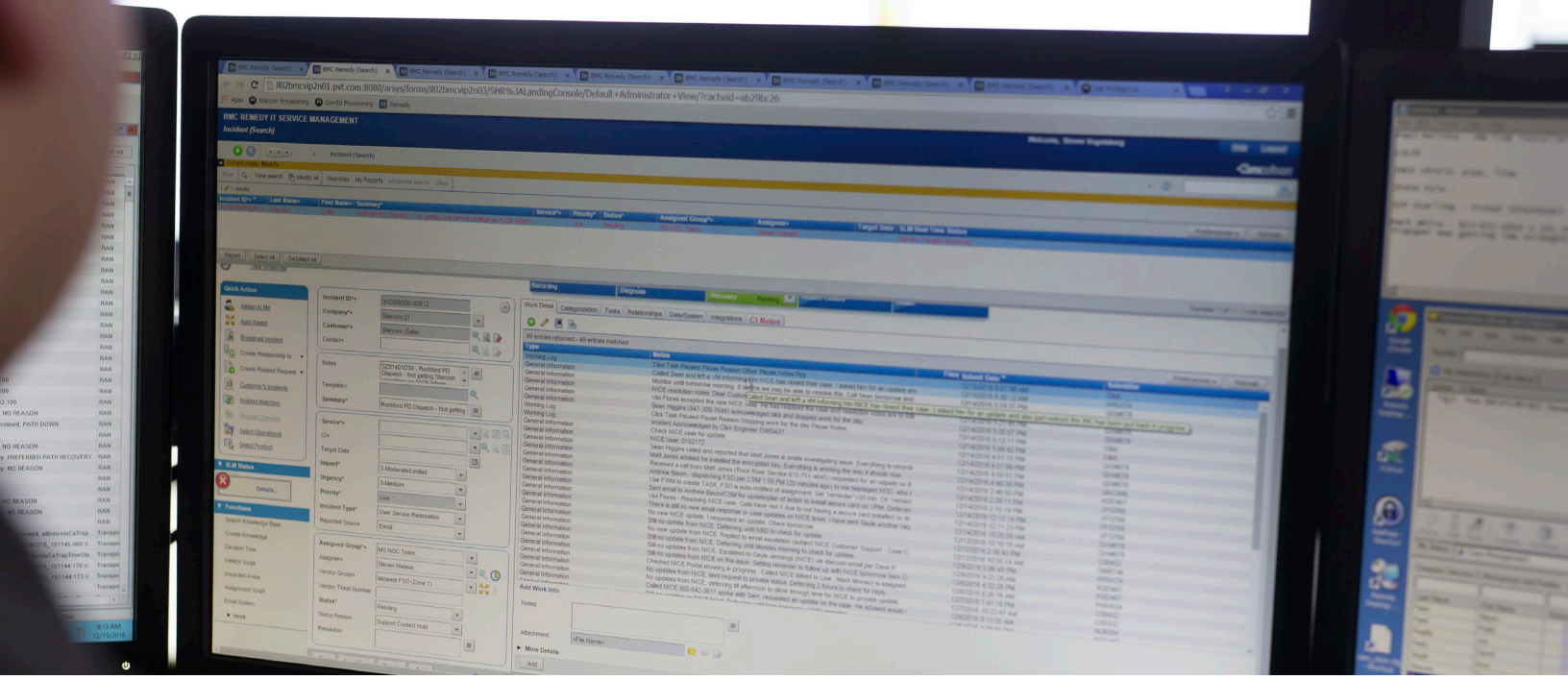
Once we're on the administrator's workstation, we often find that admins write down their passwords in an insecure format. Oftentimes it is as simple as a clear text file on their desktop, or a home share named something like passwords.txt, secret.docx, or stuff.xlsx. In addition, we sometimes find administrators storing all passwords in an encrypted file, such as a KeePass file.

On the surface, this is a good practice that we encourage of our clients. However, it is very important that this file is protected by a very strong password that only the owner knows. On multiple occasions, we have seen admins shortcut this process by saving both the location of the KeePass file and its associated password in an insecure medium (e.g., passwords.txt).

These clear text password files provide invaluable information for our testers. First, they almost always provide valid credentials to the sensitive system that we are attempting to access. It is likely possible that we could get the same credentials by starting a keylogger on the admin's workstation, but having access to the clear text file can save us a lot of time. Next, admins are typically responsible for administering multiple systems, so these password files also provide credentials to other sensitive systems we may not have discovered.

Frequently, these password files also tell us the location (e.g., hostname, IP address, web URL) of the sensitive system where the credentials are used. Again, this is something that we might have been able to identify provided enough time, but obtaining files that give us an exact roadmap of where sensitive data is stored, and how to access it, will save our testers a significant amount of time – or again, make it easier for attackers to get into this valuable information. Proper auditing of internal file shares is a very important task for any organization.

It is critical that organizations apply the principle of least privilege when assigning access controls to internal file share access. Proper auditing may also help to discover hosts that are misconfigured to allow excessive sharing. Auditing of individual stored files can help limit exposure of credentials stored insecurely within the network. Enabling two-factor authentication where possible can also help mitigate risk associated with possible insecure administrative credentials. Lastly, restricting the IP space that can log into these servers can slow down our testers from using credentials we find during the assessment.



LESSON #4 - INSIDE KERBEROASTING: CRACKING WEAK NETWORK SERVICE ACCOUNT PASSWORDS

We have demonstrated how important it is for penetration testers to get credentials that grant administrative access over hosts within the organization to escalate their permissions. Next, we will discuss a relatively recent privilege escalation technique known as Kerberoasting, which pen testers and malicious hackers can use to crack weak network service account passwords.[5]

Kerberoasting, released at DerbyCon 2014, has become a go-to technique for domain privilege escalation after gaining initial access. It takes advantage of a little-known feature of Microsoft Kerberos that allows domain users to request an encrypted version of specific service account passwords. If the passwords aren't secure, pen testers can get access to them offline using a GPU-based password cracker. There are almost always service accounts that exist in Windows domains, because there are actions that must be automated and require a service account to do its job, for example running sql accounts for your domains, or installing printers. Typically, when these service accounts are initially created, their passwords are set by a person and often forgotten if there is no prompt for these outdated passwords to be updated.

TAKING ADVANTAGE OF KERBEROASTING

To take advantage of this feature, a service account must be associated with a Service Principal Name (SPN) in the domain.[6] Most SPNs are associated with computer accounts, which by default have strong randomly generated passwords that are automatically changed monthly. Service accounts, on the other hand, are user accounts designed for a specific purpose, so the password is usually set manually by an IT administrator.[7] Whenever a password is set manually, there's the potential for it to be easy to remember, and in turn, a potential vulnerability for pen testers or attackers to exploit.

Even though it is not common to associate service accounts with SPNs, we find that most large organizations tend to have several service accounts configured in this manner. Most commonly, we find Microsoft SQL (MS SQL) service accounts configured this way. A service account generally needs to have at least local administrative access over its associated host. However, many organizations find it easier to provide the service account with access to multiple hosts.

For example, if we find a MS SQL service account called "sqlAdmin," there is a good chance that it has administrative privileges over all MS SQL servers in the organization. Occasionally, we even find that a service account has been granted excessive domain permissions, such as being placed in the Domain Admins group.

Furthermore, we find that service accounts are frequently configured once and then left alone for extended periods of time. Most service accounts that we compromise using this method were created several years ago and configured in a way that the password never expires.

HOW KERBEROASTING WORKS

Kerberoasting does not involve abusing an unpatched vulnerability in Windows Domains; instead, it leverages normal Kerberos authentication.[8] In short, the way Kerberos allows users to authenticate to domain services (file servers, MS SQL servers) is through Ticket-Granting Service (TGS) tickets.[9] A portion of this TGS ticket is symmetrically encrypted with the password hash of the account for that service.[10] For example, if a user wants to access a share, or a file on a file server named FS01, the user will request a TGS ticket for the CIFS service for host FS01.[11]

This ticket contains all the information that the file server FS01 will need to validate if the user has access to the CIFS service. To prevent the user from modifying the authentication data in the TGS ticket, the data is symmetrically encrypted with the NTLM password hash for the account associated with this host's CIFS service. By default, that would be the NTLM password hash for the file server's computer account "FS01$." Pen testers can request this TGS ticket and try to brute force the account password. However, by default, the computer account password is a randomly generated powerful password that is changed monthly, so it would be next to impossible for us to break through. Even if it were to be cracked, the password would only be valid for the remainder of that month.

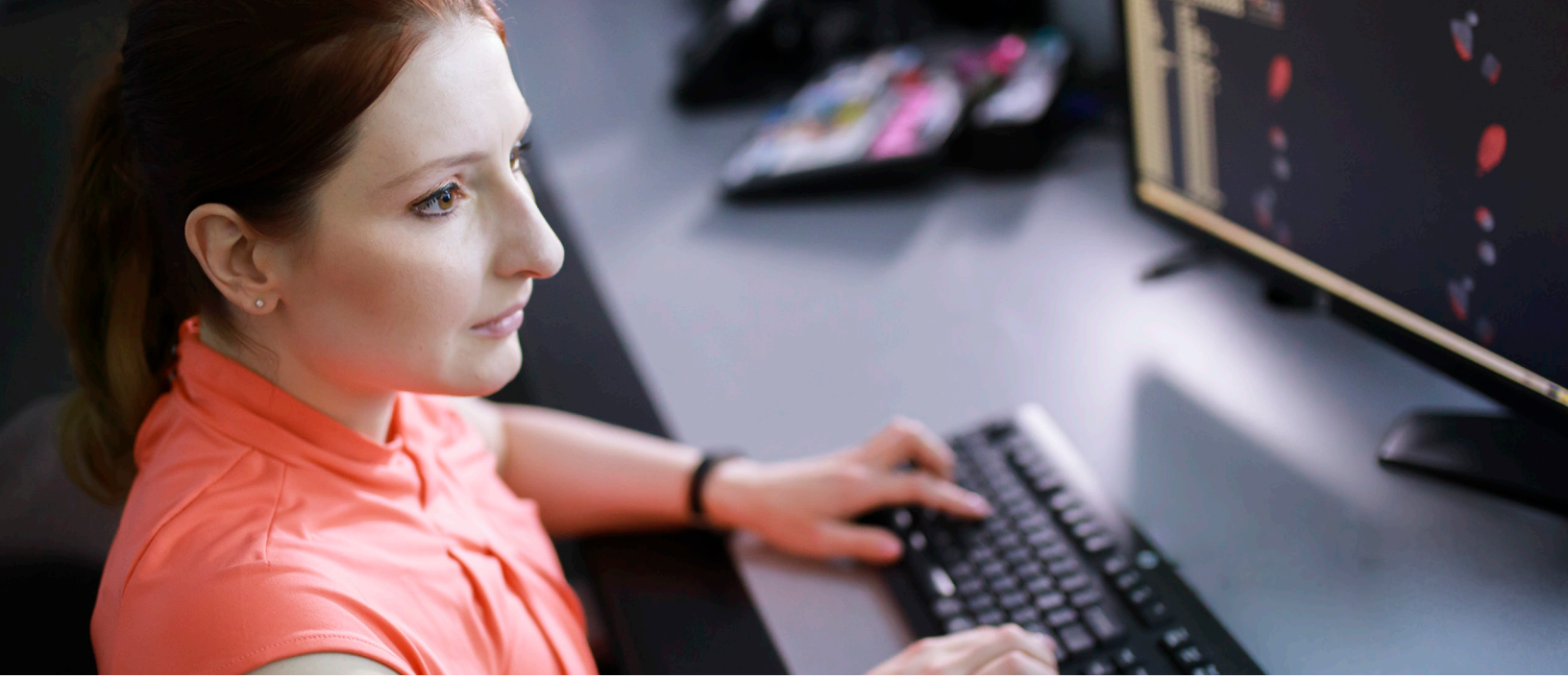
Weak passwords that can be cracked quickly are the root cause for Kerberoasting effectiveness. By having and enforcing a complex password policy for all users, especially for service accounts associated with SPNs, an organization can defend against these attacks.

Not all services in the domain are associated with computer accounts; it's highly dependent on the organization. But our operators can usually find at least one SPN that has been associated with a non-computer account. We go through the same procedure as before, but this time we convert the TGS ticket into a hashcatcompatible format that we can try to crack offline.

Another feature of this technique is that only internal network communication can take place between our compromised host and the Key Distribution Center (KDC), which is typically a Domain Controller.[12] Requesting of TGS tickets is typical Windows domain traffic, and no traffic is sent to the host running the service we are trying to access. This enables us to request these tickets for any service in the forest, not just our current domain. This is important because even if firewall rules prohibit lateral movement into another domain, we can still request TGS tickets that are encrypted with the password hash of service accounts in another domain.

Note that this technique is highly dependent on the service account using a weak password that we can crack during an assessment. Many organizations believe that the primary threat to weak service account passwords is an attacker attempting an online brute-force attack. In this case, the attacker would have a limited number of guesses before the account is locked out and security staff is alerted. Kerberoasting provides an avenue for the tester or attacker to conduct an offline brute-force attack. Many passwords guesses can be attempted in a short timeframe without the possibility of an account lockout. For instance, a single modern consumer-grade GPU can guess well over 100 million passwords per second.

Weak passwords that can be cracked quickly are the root cause for Kerberoasting effectiveness. By having and enforcing a complex password (12 characters or more with a mix of numbers/letters/specials/upper/lower) policy for all users, especially for service accounts associated with SPNs, an organization can defend against these attacks. One way to mitigate some of the effectiveness of Kerberoasting is by using Microsoft's "Managed Service accounts."[13] Once set up, Managed Service accounts will automatically change passwords for these accounts every month, thus lowering the usefulness of any cracked passwords to a month at most.



LESSON #5 - HOW OUR PEN TESTERS GET THROUGH MAIL APPLIANCES

To wrap up our lessons learned from security assessments, we'll discuss a high-risk issue our penetration testers and consultants often come across: filtering malicious emails. In our assessments, sending phishing emails with malicious payloads or links is the most common method we use to get initial access to a network. Email is one of the few vectors that gives us direct access to end users. Overall, there is a much higher chance a single user will unwittingly click on a malicious link or open a harmful attachment than fall for other methods we use to try to compromise them. However, this is predicated on those malicious emails successfully making it to the end user's inbox.

Many security teams believe their email filtering appliances are the first line of defense, and the user is the last line of defense when it comes to stopping phishing. In our experience, security teams end up relying on end users to be the "catch-all" for malicious emails, instead of attempting to make it more difficult for a malicious email to make it to a user in the first place.

BASELINE PROTECTION: SCANS AND SANDBOXING

Most commercial grade email appliances offer some basic scanning or sandboxing to make sure emails are "safe enough" to be delivered to end users. However, skilled attackers are always prodding and testing to find ways to bypass these security measures and get malicious payloads, or links to malicious payloads, past them. As a first step, a well-trained security team should know the capabilities and gaps of its mail server defenses. By examining your organization's own defensive capabilities, you can identify and fix weak spots. This means knowing what steps your mail appliance is taking to analyze emails, and how it decides to deliver them to end users or not.

One cause of malicious payloads ending up in users' inboxes stems from a sandboxing device failing to classify the payload as malicious. Specifically, devices run the payloads in a "safe" environment to see what they do (i.e., call out to domains, attempt to download files, attempt to inject code into other processes). The classification process is usually when security professionals have the least visibility into how their appliance is working. This is an area you should look at closely when evaluating your network defenses. In our experience, sandboxes will run a minimal number of tests to determine if the attachment is considered malicious. In addition, security researchers have identified malware that is specifically designed to recognize when it's in a sandbox environment and thus evade detection.

EVADING MAIL SERVER DEFENSES

One technique we typically use before we begin our pen tests is to register a few domain names that sound benign, such as "federalbusinessnetwork.com," so that we can send our emails from a legitimate account, such as hr@federalbusinessnetwork.com. We start by being a reputable domain name, fooling mail servers into trusting it.

Given enough time, we can typically find ways around tests and trick a sandbox into categorizing our payload as benign. For example, some sandboxes may only run a payload for a short period of time, so they can be bypassed by simply adding a "sleep" statement into the beginning of our code.

One of our preferred methods – and of threat actors – is delivering malicious payloads to an end-user by attaching a malicious MS Office document with an embedded Object-Link Embedded (OLE) payload.[14] The "OLE" payload has become a very common phishing technique. While some email appliances detect the presence of the OLE objects, and can inspect them to see if they're malicious, not all are blocked at the mail appliance and are thus delivered to end users.

OFFICE OF HUMAN RESOURCES
Example Company
Washington, DC

Feb 23, 2017

MEMORANDUM FOR COMPANY EMPLOYEES

FROM: John Smith
Director of HR and Administration

TO: ALL PERSONNEL

SUBJECT: Social Media Policy Update

We use social media websites to engage in dialog that increases government transparency, promotes public participation, and encourages collaboration with the public. We currently maintain official accounts on the following social media websites: Facebook, Twitter, YouTube, and Flickr.

When you visit a company page on any of the above social media sites, the privacy and security policies of those websites apply. Our Company does not control, moderate, or endorse the comments or opinions provided by visitors on these sites. These social media sites each have their own privacy notices and we encourage you to read each policy for the social media sites that you use.

It is critical for employees to understand how to handle the upcoming diversity of social media as it affects how the workforce interacts with the public.

Please read and refresh yourself with some examples of improper public interaction on social media sites:

Figure 3: The end user experience of opening the attached MS Office document might look something like this.

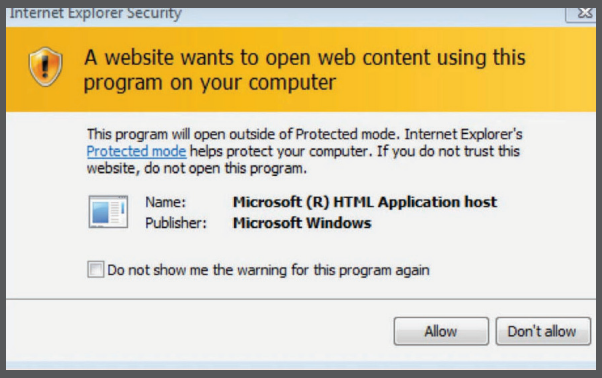


Figure 4: Warning about executing code from the HTA file

The end user experience of opening the attached MS Office document might look something like figure 3.

The embedded file at the bottom is disguised to look like a .docx file, but is actually a batch file that will run code of our choosing when the employee double-clicks the icon. Frequently, the script or object is surrounded by text that encourages the user to click or interact with it.[15]

Although the malicious Word document with an embedded OLE object works for our assessments, it requires several clicks from the user before the payload is executed. Another option we have is sending the user an email containing a link to an HTML Application (.hta file). We configure the HTA file to execute malicious vbscript code. If the user clicks on our link using either Chrome or Firefox, this file will be downloaded and the user must manually open it to run our payload. However, if the user opens our link using Internet Explorer, which many organizations still use, the user will be immediately prompted to open the file. If the file is clicked, the user will be presented with a warning about executing code from the HTA file.

We find that many users click past this warning, which grants us control of their workstation in the user context of whoever clicked the link. Many commercial and open source penetration testing tools are now incorporating HTA creation as part of their attack flow such as Metasploit[16] and Powershell Empire.[17]

Our pen testers have been using both approaches for the past few years, and we're just starting to see some organizations detecting or blocking these payloads from being delivered to users. The situations in which the payload makes it through is always a result of the defensive security stack failing to prevent the blocking of the email and its payload.

A good approach to defending such attacks is to have insight into how your security department filters and scans for malicious emails and potential vulnerabilities. Have your internal team or an outside organization try to get through your mail defenses by sending malicious emails to test the capabilities and limitations of your mail defenses. If you're using a third-party solution for email filtering, make every effort to have some level of recourse with your vendor to submit suspected malware samples for manual review when needed. And as always, be sure your users are trained to recognize, report, and not open any suspicious emails nor click on any links or attachments within them, even if they appear to come from a legitimate source.

CONCLUSION

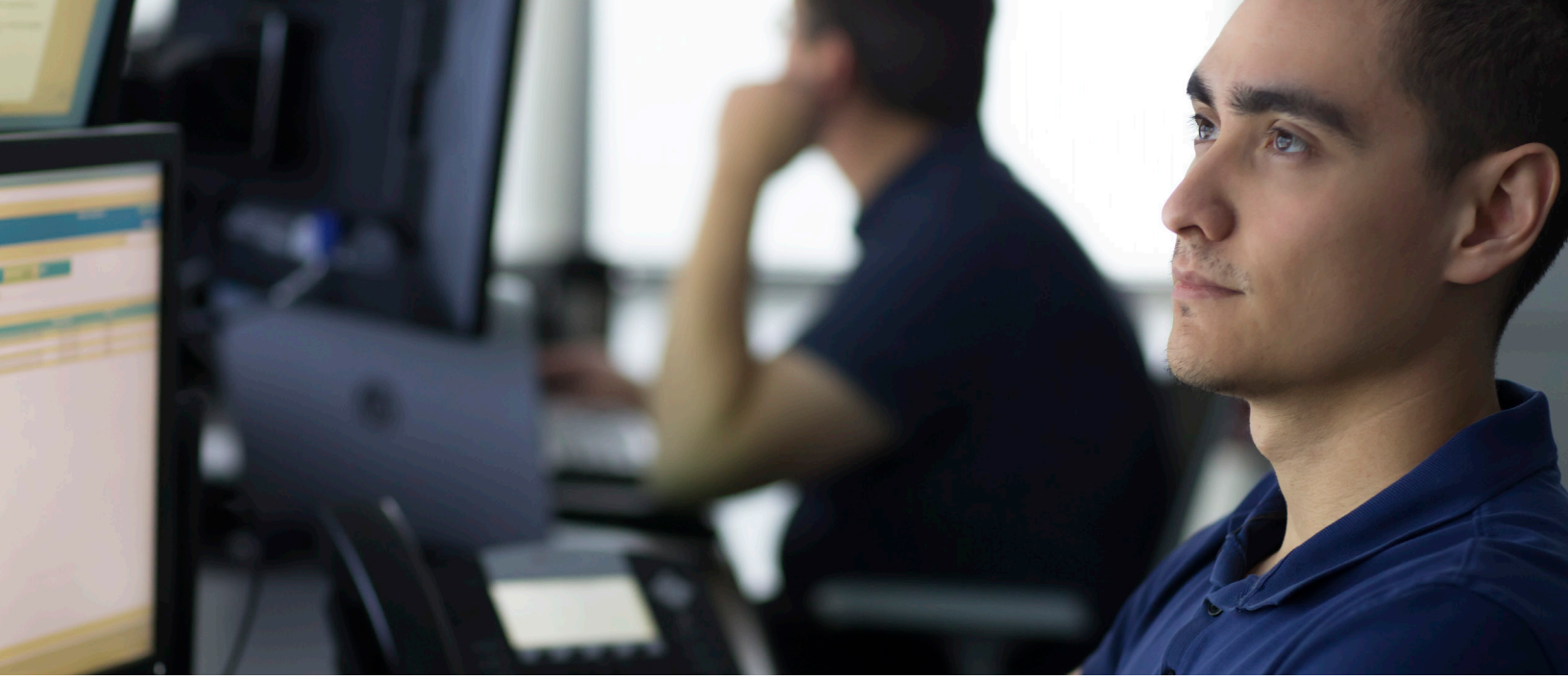
These lessons learned only touch on our top five findings, but they are among the most common ones we have found in the past. Although penetration testing is often seen as a means to simply "check the box" for compliance, a thorough assessment that includes external, internal and wireless networks is critical for ensuring organizations properly understand their risks and vulnerabilities. Organizations that are serious about defending against threats should consider regular assessments as an essential part of their cybersecurity program, and work with a vendor that can create and deliver a customized approach specific to their business needs.

THE MOTOROLA SOLUTIONS PENETRATION TESTING APPROACH

Our pen testers take a direct, simulated attack approach against some of the toughest security defenses.

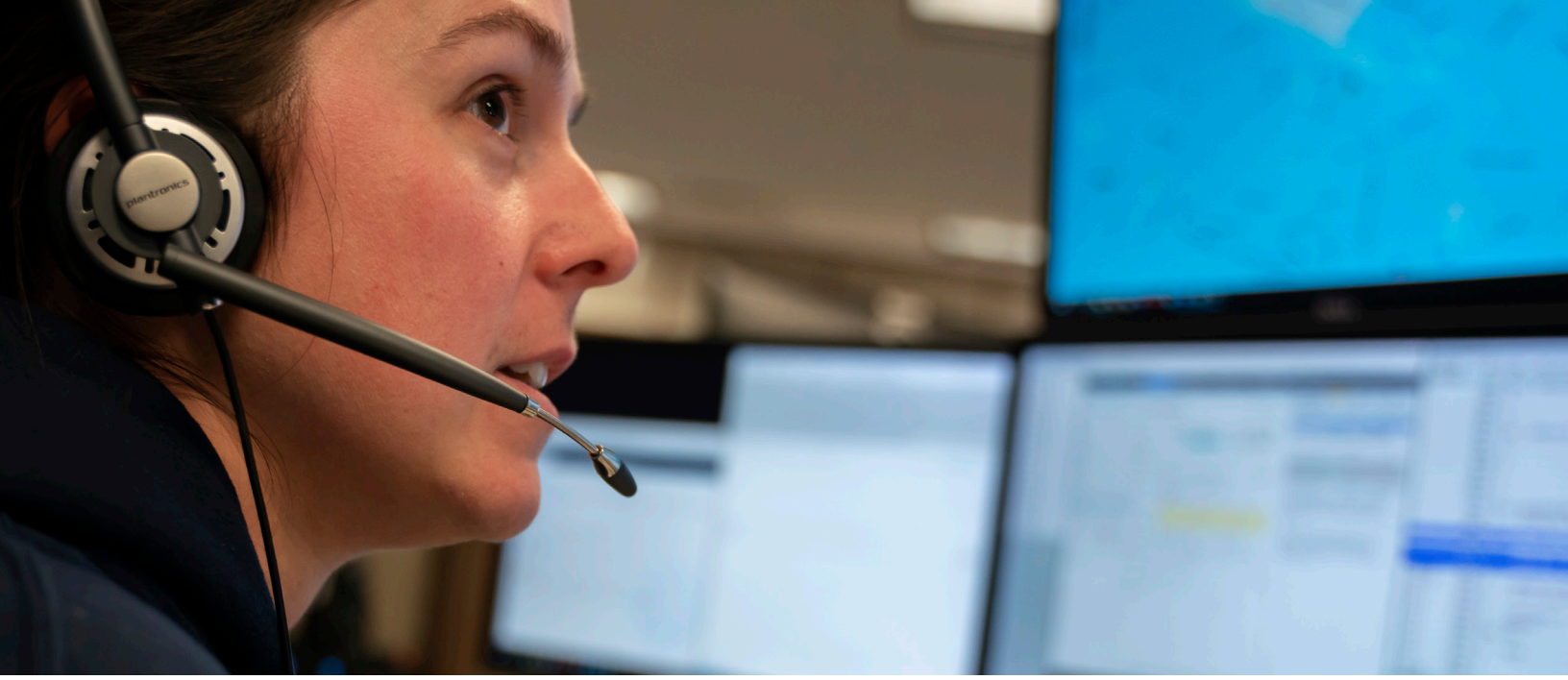
Our penetration tests assess the effectiveness of security operations people, processes and technology. Part one is an external engagement, and part two is an internal engagement. The external portion is conducted remotely and replicates an attack path that a geographically separated adversary might take. Unless the client specifies otherwise, the test is conducted "black-box" style, where the testers have minimal knowledge of the organization beforehand.

Our goal is to mimic adversaries as closely as possible while remaining tool-agnostic. We simply use the best tools and techniques for each job. The technical objective of our assessments is to emulate an outside adversary to get access into an internal network, escalate privileges and obtain sensitive information. The intent is not to find every single vulnerability in the way that a vulnerability scan might do, but rather to find some of the vulnerabilities that exist, and attempt to exploit those. To an executive, the output from your favorite vulnerability scanner is not as compelling as an actual attack path that someone can take advantage of in the real world. We are proponents of the "assumed breach" mindset; for everyone to get better, it is best to assume you have been breached and try to figure out how to detect and respond after an attack has happened.



ENDNOTES

1. DLL Hijacking Attacks Revisited, InfoSec Institute: http://resources.infosecinstitute.com/dll-hijacking-attacks-revisited/#gref
2. GitHub: https://github.com/BloodHoundAD/BloodHound
3. Clear Text Definition, TechTarget: http://whatis.techtarget.com/definition/cleartext
4. GitHub: https://github.com/PowerShellMafia/PowerSploit/blob/master/Recon/PowerView.ps1
5. Tim Medin, Attacking Kerberos: Kicking the Guard Dog of Hades, SANS Summit: https://youtu.be/LmbP-XD1SC8
6. Service Principal Names, Microsoft: https://msdn.microsoft.com/en-us/library/ms677949(v=vs.85).aspx
7. Service Accounts Overview, Microsoft, TechNet: https://technet.microsoft.com/en-us/library/dn617203(v=ws.11).aspx
8. Kerberos Definition, TechTarget: http://searchsecurity.techtarget.com/definition/Kerberos
9. Ticket-Granting Tickets, Microsoft: https://msdn.microsoft.com/en-us/library/windows/desktop/aa380510(v=vs.85).aspx
10. Andy Greenberg, Hacker Lexicon: What is Password Hashing, Wired: https://www.wired.com/2016/06/hacker-lexicon-password-hashing/
11. Common Internet File System, Microsoft: https://technet.microsoft.com/en-us/library/cc939973.aspx
12. What is Key Distribution Center (KDC), IGI Global: https://www.igi-global.com/dictionary/key-distribution-center-kdc/16150
13. Microsoft, TechNet: https://technet.microsoft.com/en-us/library/dd560633(v=ws.10).aspx
14. Object Linking and Embedding, Wikipedia: https://en.wikipedia.org/wiki/Object_Linking_and_Embedding
15. Where's the Macro?, Microsoft TechNet: https://www.microsoft.com/security/blog/2016/06/14/wheres-the-macro-malware-author-are-now-using-ole-embedding-to-deliver-malicious-files/
16. Rapid 7: https://www.rapid7.com/db/modules/exploit/windows/misc/hta_server
17. GitHub: https://github.com/EmpireProject/Empire



BRIDGE THE GAP TO A SECURE CLOUD

Whether you're migrating to the cloud or already there, Motorola Solutions can provide the expertise and solutions you need to keep your organization secure. Explore our other resources to learn more.

Motorola Solutions delivers cloud security, SOC-as-a-Service, managed security, and professional services to commercial and public sector clients. We provide the visibility and control needed for effective cloud, endpoint, and network security to bridge the gap to a modern security approach. ActiveEye℠, our proprietary platform, uses Security Orchestration Automation and Response (SOAR) to optimize and scale Managed Detection and Response (MDR) capabilities across the enterprise. Our US-based cybersecurity experts provide 24/7 monitoring, consulting, and guidance to our customers on their journey to a secure environment. Professional services include penetration testing, exercises and training, vulnerability assessments, threat hunting, and incident response.

Learn more at: www.motorolasolutions.com/cybersecurity

MOTOROLA SOLUTIONS

Motorola Solutions, Inc. 500 West Monroe Street, Chicago, IL 60661 U.S.A. **motorolasolutions.com**