RAPID7

UNDER THE HOODIE:

Actionable Research from Penetration Testing Engagements

By Bob Rudis, Chief Security Data Scientist, Rapid7, Inc.

Tod Beardsley, Research Director, Rapid7, Inc.

Andrew Whitaker, Director, Global Services, Rapid7, Inc.

February 8, 2017

CONTENTS

EXECUTIVE SUMMARY

This paper seeks to demystify the practice of penetration testing by surveying the engineers and investigators themselves on what they most commonly see during engagements with clients. We have taken the results of 128 penetration tests, which were conducted by Rapid7's penetration testing team primarily in the fourth quarter of 2016, and synthesized the data collected from those engagements in order to present an accurate depiction of what any IT operations manager can expect from their next (or first!) penetration test[1].

Most penetration testing clients are more interested in external, rather than internal, penetration tests[2], and they are concerned with assessing the security of their customers' and employees' personally identifiable information rather than trade secrets or source code. This is true across all sampled industries[3].

Regardless of scope or industry, a typical penetration testing engagement tends to successfully identify and exploit software vulnerabilities about two-thirds of the time. In addition, network misconfigurations are discovered and leveraged for inappropriate access at approximately the same rate (about 67% of the time), again regardless of scope or industry. Overall, penetration testers successfully compromised the target organization through software vulnerabilities or network misconfigurations just over 80% of the time. After all, most organizations—large or small, across every vertical—have a network infrastructure built from commonly available software and hardware, leading to a homogenous, cross-industry environment dominated by large and popular software distributors, both proprietary and open source. All of these technology suppliers produce software with vulnerabilities and tend to favor deployability and usability over security, and as a result, the problems that are common in one company's network tend to be problems in all company networks.

The good news is that, precisely because of this homogenous environment, most of the techniques used can be defended against with sensible, widely understood and appropriately tailored network security best practices, including patch management, network segmentation, and regular assessments of the most likely sources of risk in the enterprise.

Compromised credentials should be thought of as both a means and an end, and they are among the most difficult features of an enterprise network to defend. During an engagement, credentials are successfully compromised nearly half of the time. Of those, approximately half can be used to escalate privilege to site-wide administrators. The number one method of obtaining account access starts with very simple password guessing; enforcing more machine-generated, rather than human-generated, passwords would go a long way toward defending against this threat, as would more widespread adoption of two-factor authentication[4].

Taken together, credentials were compromised, misconfigured network features were leveraged, or software vulnerabilities were exploited on 86% of all engagements performed over the census period, and we believe this is a fairly typical success rate for an average penetration test. We also found that neither enterprise size nor business vertical appear to matter to the penetration success rate or the rate at which penetration testers were "caught" during testing by detection controls.

Most penetration testing clients are more interested in external, rather than internal, penetration tests

[1] Note, this is a representative sample from Rapid7 engagements but does not constitute the entire population of all of Rapid7's engagements for the observed quarter.

[2] During the sampled period, external web application assessments made up 31.3%; external network tests, 24%; internal assessments, 19%; social engineering, 4.2%; and wireless infrastructure, 2.4%, with the remaining being custom engagements such as source code analysis, IoT product design, etc.

[3] The industries covered during the sampled period were services (23%), finance (20%), communications (13%), healthcare (9%), utilities/energy (9%), tech (8%), retail (7%), manufacturing (5%), other (3%), and transportation, real estate, and education (1% each).

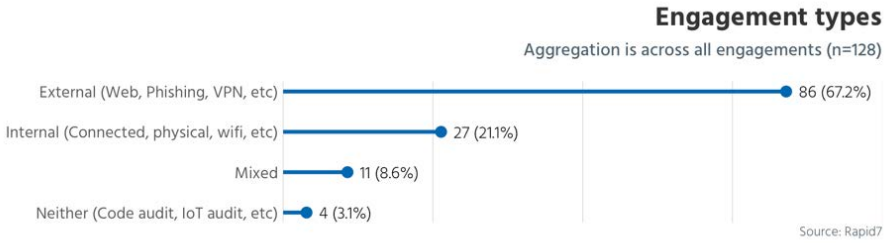
[4] Humans really aren't very clever with passwords. The most common login password formula seen on engagement is "<Season><year>!" since "Winter2017!" nearly always satisfies most corporate password complexity rules, and can be rotated every 90 days without effort.

DETERMINING A PENETRATION TEST SCOPE

The first order of business for any penetration tester is to get a sense of the scope of an engagement, since the activities involved are going to vary depending on a variety of factors.

Internal or External?

For engagement scope, we broadly categorize engagements as either internal, external, mixed, or neither[5]. We found that most client organizations are more concerned about their externally facing network assets than internal; of all the engagements surveyed, 75.8% are either entirely or partially concerned with external exposure.



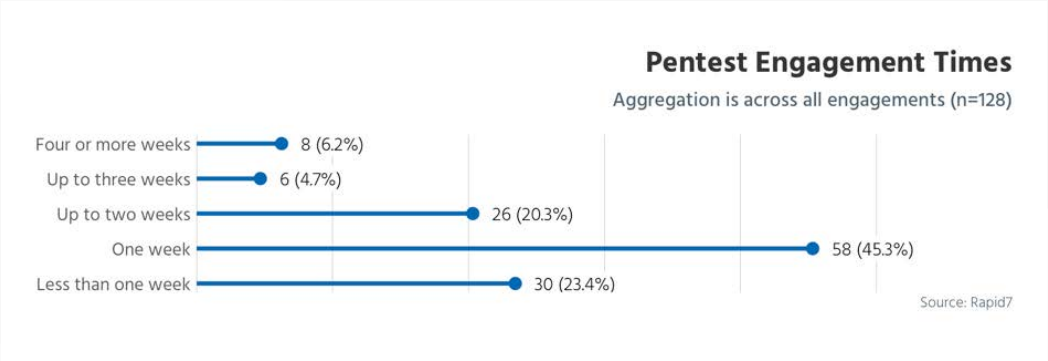
This preponderance of external assessments isn't a particularly surprising finding since the vast majority of criminal activity is externally, and often internationally, based. Plus, external assessments tend to be the least common denominator across virtually all compliance and regulatory frameworks. Phishing assessments, where a company's employees are sent emails designed to trick them out of their credentials or to install custom malware, are increasingly common components of a complete penetration test. Also encouraging is the finding that one in five of the penetration tests surveyed was entirely internal in scope. Taken together, these data points speak to the growing recognition that a defense-in-depth philosophy of network security is ultimately what's required in the face of an increasingly hostile internet, and organizations are seeing value in securing their network assets—and internet habits—beyond the bare minimum standards that are enforced by industry compliance rules.

The Time Box

The length of an engagement is an especially important constraint on a penetration tester; most penetration testers will lament the amount of time contracted, candidly stating that "real threat actors" have all the time in the world to breach a network. It's true that a dedicated, focused, human-driven attack campaign may spend weeks to months performing reconnaissance and may remain undetected for weeks or months after a successful breach, and most security vendor marketing material will say as much. While all internet-connected organizations bear some risk from targeted, willful attackers, the vast majority of individual "attacks" by volume are unsophisticated, automated botnets that spend mere seconds on a particular target before moving on, seeking out the lowest of low hanging fruit.

[5] The "neither" category would be an engagement like source code analysis, an IoT product design review, or any other engagement that doesn't involve any "live" targets, internal or external.

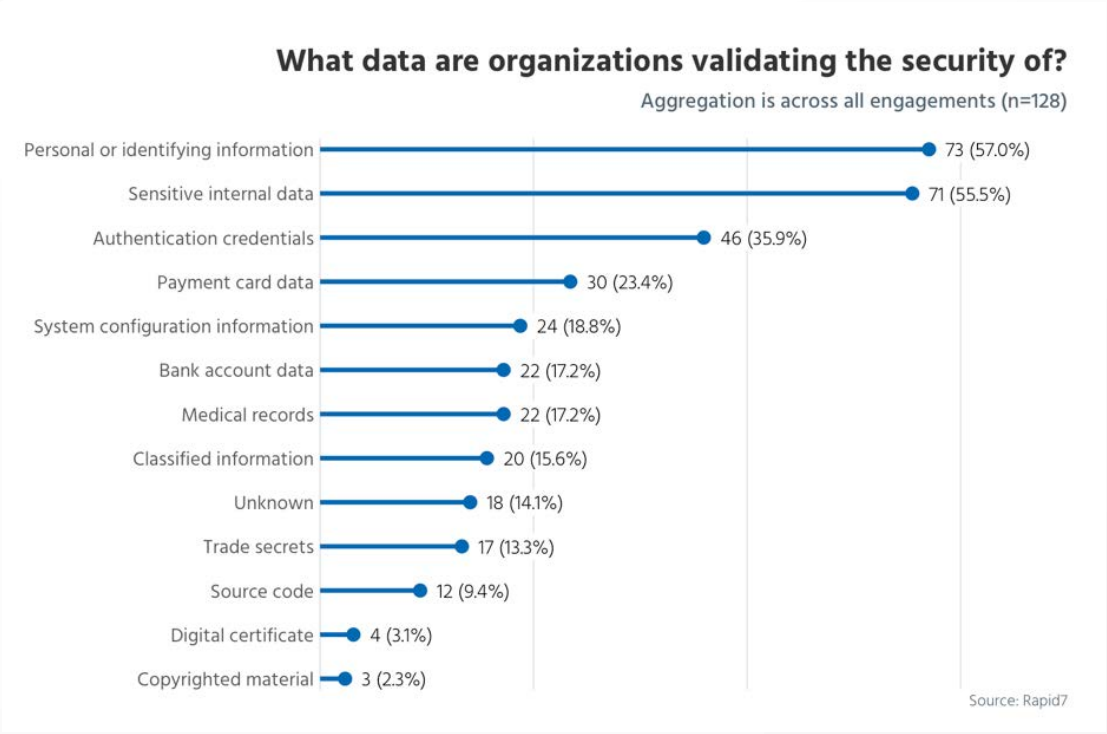
Because of these extremes, and the fact that no attacker has truly unlimited time, a reasonably time-boxed engagement should be effective in unearthing the kinds of issues the client is interested in mitigating. Of course, penetration testing doesn't come for free, and the budget available to hire penetration testers for an engagement is almost always the limiter for the amount of time the penetration tester has with a given client. We can see from the graph below that most clients—over two-thirds—budget one week or less for a typical engagement, while only 6.2% expect four or more weeks of effort.



What's at Stake?

Finally, the object of the penetration test is a critical factor in determining the penetration tester's strategy. When organizations contract for penetration testing, rarely will they simply say "hit me with all you've got!" Instead, most clients are interested in protecting specific kinds of data or networked assets. Personally identifiable information (PII) tops the list of concerns for penetration testing clients, followed by the somewhat nebulous "Sensitive internal data," which can be anything from the content of internal communications to undisclosed financial metrics.

And yet, despite the recent uptick in online industrial espionage[6], the surveyed organizations seemed the least interested in specifically protecting copyrighted material, digital certificates, source code, or trade secrets. These data types are certainly valuable, even critical, to the continued viability and success of an organization, but the pressure to protect PII and PII-like data is clearly apparent in the chart below[7].



[6] As reported by the Verizon 2016 DBIR.

[7] Other constraints on scope, such as the known fragility of certain factory floor assets, or the dire consequences related to accidentally knocking life-saving medical gear offline, are almost certain to contribute to favoring assessing PII protections over operational assets.

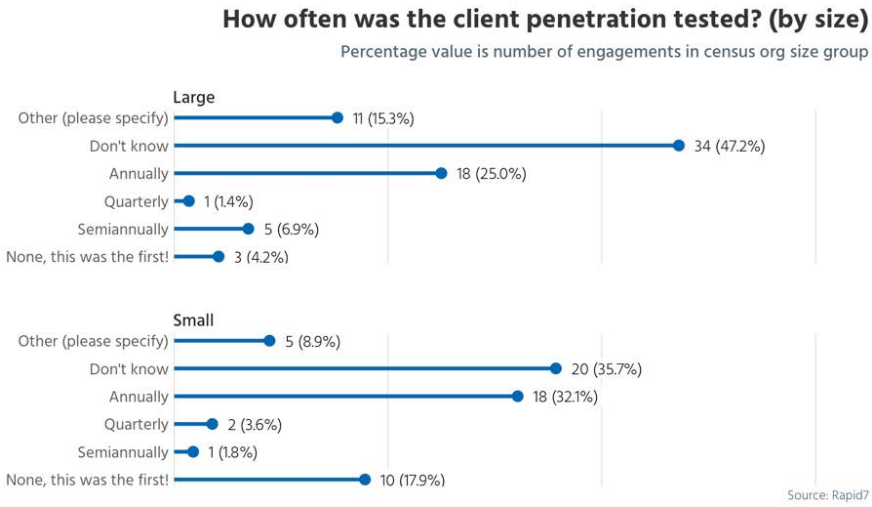
TARGET DEMOGRAPHICS

For demographic data, our survey captured the relative sizes of the organizations that contracted Rapid7 for penetration testing, what businesses they're involved in, and how experienced the targets are in handling penetration tests. We can draw some conclusions on the differences between organizations based on these points of demographic data.

Target Size

We found that small organizations—those with fewer than 1,000 employees—are more likely to be brand new to penetration testing. This isn't all that surprising, as smaller companies tend to be newer and are only now subject to regulatory and compliance standard mandated penetration tests.

The figure below details the frequencies at which the 72 large organizations and 56 small organizations in our sample have penetration services conducted.



For both large and small organizations, it's clear that there's a preference for annual penetration tests; this is certainly due to compliance and regulatory requirements that mandate this minimum level of penetration testing. On the one hand, an annual, fourth-quarter penetration test is certainly better than nothing. On the other, the pace at which enterprise networks are changing is only increasing, and it seems unlikely that an annual penetration test is sufficient to catch the cumulative changes that occur on a quarterly or semiannual basis.

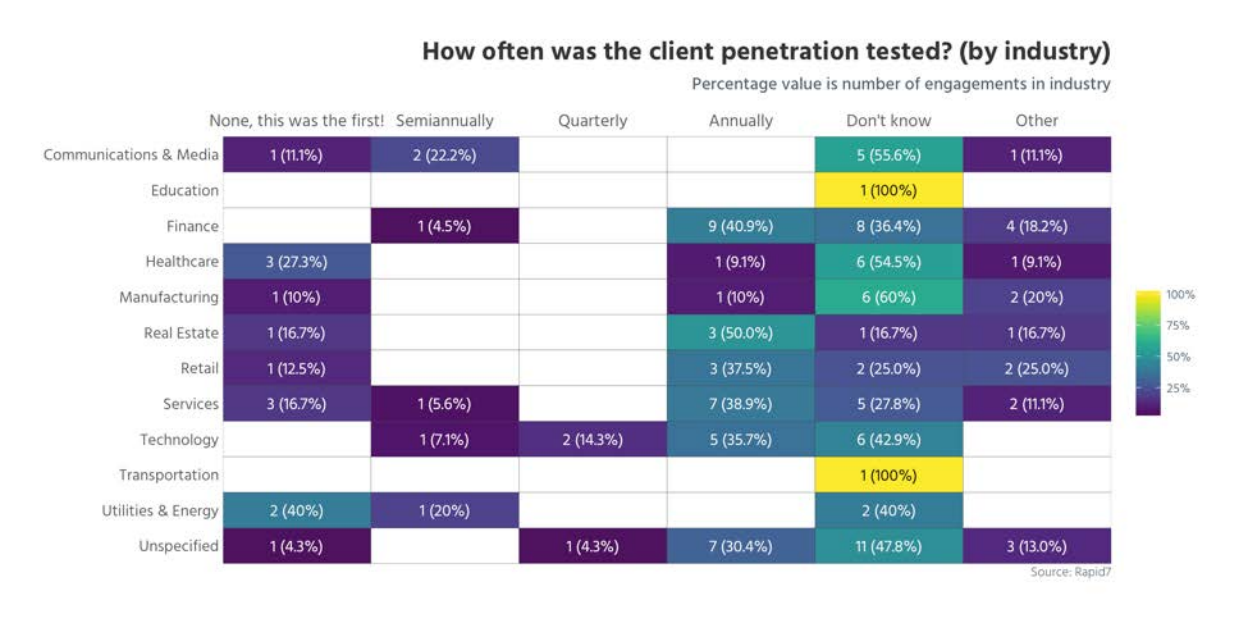
Both large and small organizations tend to favor external penetration tests, with a slightly stronger preference for external among smaller organizations. Small organizations sign up for external penetration tests about 70% of the time, while larger organizations are closer to 60%. We also saw essentially no difference in engagement durations between large and small organizations; both favor one to two weeks.

Get more "Under the Hoodie" insights at:

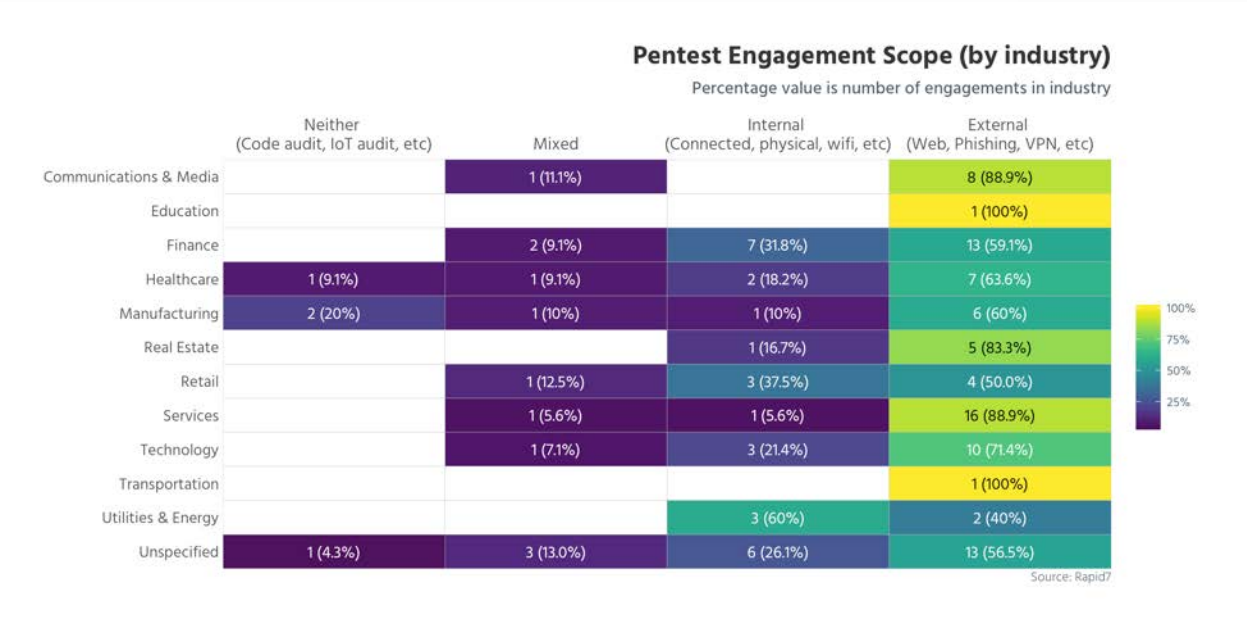
www.rapid7.com/info/under-the-hoodie

Target Industry

We were somewhat surprised to find very few differences between surveyed clients when we factor in their respective verticals. We expected to find that the highly regulated financial and healthcare sector businesses would be consistent among one another when it comes to their experience with penetration testing. However, no vertical expressed a majority-predictable penetration testing schedule; the most consistent vertical surveyed was real estate, with 50% of these clients conducting annual penetration tests, whereas only 41% of finance-oriented businesses have a clear annual penetration testing policy. Many organizations, regardless of sector, either did not disclose their typical penetration testing frequencies or this was their first engagement.



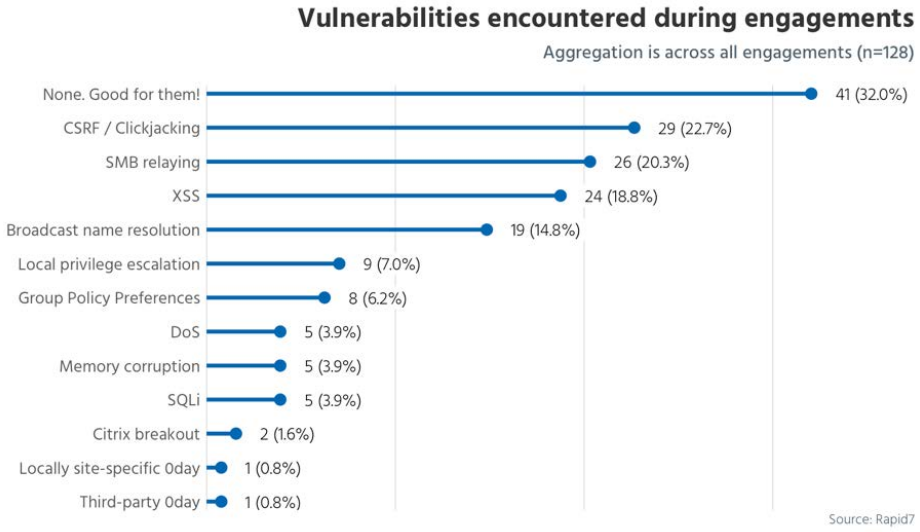
While all industries favor external penetration tests over internal, we did find that financial and retail businesses opt for internal penetration testing more often than the cross-industry norm of 21%. This indicates that finance and retail businesses, perennial favorite targets for PII-seeking criminals, are taking serious steps to help shore up their internal network defenses.



VULNERABILITIES ENCOUNTERED

A software vulnerability, at its core, can be thought of as an undocumented, usually unintentional, application program interface (API). They're nearly always created by accident (as a bug), and their associated exploits are merely specialized programs that take advantage of these APIs to perform actions that are unexpected by both the software designer and the software user.

As unfortunate as software vulnerabilities are, we appear to be stuck with them; it is currently impossible to ship software of any reasonable complexity without bugs, and sometimes, these bugs lead to vulnerabilities. On penetration testing engagements, just under a third of sites tested appeared to be free of exploitable vulnerabilities.



This leaves over two-thirds of client sites with some sort of vulnerability to be exercised. Exploiting vulnerabilities remains the most common activity people think of when imagining what penetration testers do on engagements, and indeed, finding and exploiting vulnerabilities on a target network is still fairly commonplace.

Since we know that most penetration tests are externally based, it's unsurprising to see that the common web application vulnerability classes of cross-site request forgery (CSRF, or CWE-352), clickjacking (CWE-451), and cross-site scripting (XSS, or CWE-79) top the list as the most commonly encountered vulnerabilities, because most external engagements involve a web application testing component. These vulnerability classes are all extremely common web application vulnerabilities, and they can affect any custom web application that does not treat user input with a healthy dose of suspicion. Attackers can leverage these vulnerabilities to achieve effects ranging from temporary site vandalism to total web server compromise[8].

On the internal side, we can also see that SMB relaying is the most common vulnerability encountered, and it is a technique for impersonating Windows-based clients and servers to each other. Combined with broadcast name resolution (the fourth most common vulnerability), this tends to quickly lead to total exploitation of most Microsoft-based networks[9].

Exploiting vulnerabilities remains the most common activity people think of when imagining what penetration testers do on engagements, and indeed, finding and exploiting vulnerabilities on a target network is still fairly commonplace.

[8] The Tangled Web: A Guide to Securing Modern Web Applications by Michal Zalewski, published in 2011 by No Starch Press, is the definitive reference for common web application and web browser vulnerabilities, and can be found at http://isbn.nu/1593273886.

[9] For more on SMB relay as a vulnerability and SMB signing as a partial mitigation, see Leon Johnson's excellent 7-minute video explanation at https://www.rapid7.com/resources/smb-relay-attacks-explained/ .

On the internal side, we can also see that SMB relaying is the most common vulnerability encountered...

Between the internal and external engagements, it's not terribly shocking to discover that internal engagements reveal far more vulnerabilities than the external engagements. Internal networks often have all kinds of unexpected, rarely used, and rogue services running, and even today, many devices lack basic host-based firewalls or endpoint solutions. External networks tend to offer far fewer services, and thus, have fewer chances to offer vulnerable, unpatched services.

86 External Engagements	47 revealed at least one vulnerability	44.2% vulnerability rate
27 Internal Engagements	26 revealed at least one vulnerability	96.3% vulnerability rate
11 Mixed Engagements	9 revealed at least one vulnerability	82.8% vulnerability rate

Finally, it's important to note that for the vast majority of identified vulnerabilities, these software bugs were already known by the software vendor or developer. However, the (very) occasional 0day—a software issue that is unknown to the vendor or developer—still pops up from time to time while on engagement. These are definitely black swans when it comes to penetration testing; most exploit-based attacks, real or simulated, rely on commonly known vulnerabilities surviving in networks that lack perfectly comprehensive patch management systems. That said, some proportional amount of attention and effort should go to configuring the enterprise network to at least contain the damage caused by those occasionally unpatchable, difficult to mitigate vulnerabilities.

UNDER THE HOODIE:
From Cross-Site Scripting to Total Network Compromise

Client Vertical: Healthcare

While many penetration testing clients might expect a web application assessment to merely list out the common vulnerabilities discovered during the assessment, one memorable engagement conducted by Rapid7's Matt Schmidt and Phil Bosco resulted in a far more dramatic demonstration of risk. By chaining together seemingly unrelated web application vulnerabilities, Schmidt and Bosco not only compromised the target's web application, but they went on to leverage these vulnerabilities to gain total control over the back end infrastructure.

The exploitation chain began with the discovery of a Cross-Site Request Forgery (CSRF) vulnerability in a healthcare provider's externally facing web application. When an administrative user loaded a malicious HTML page sent by the penetration testers, an unverified request was delivered to the web application to create a new account on the server, without notification or otherwise alerting the tricked administrator.

Using their newly created account to authenticate to the webapp, the pair discovered a persistent Cross-Site Scripting (XSS) vulnerability. This vulnerability was then leveraged to 'hook' any logged-in administrator's browser session into an exploitation framework, and once hooked, steal that administrator's session token and thus impersonate the administrator, gaining all of his rights and privileges within the application.

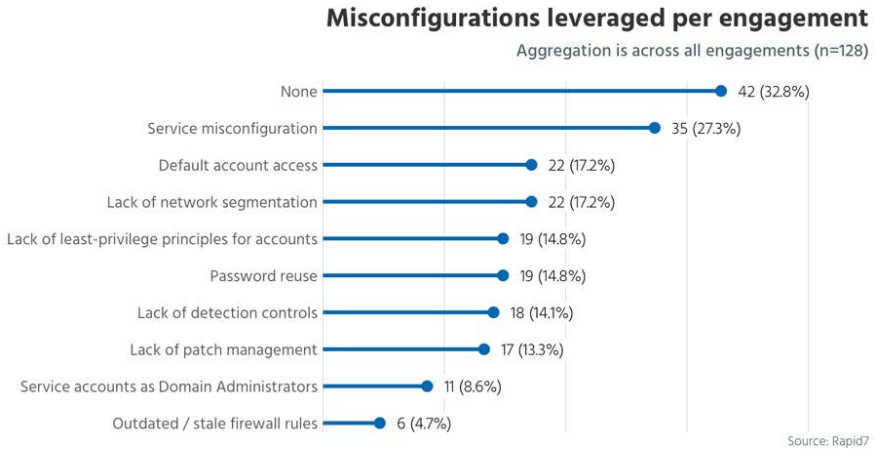
Now armed with application administrator privileges, the pen testers discovered the webapp failed to sufficiently validate uploaded file types. By altering the expected POST request to appear as if the file type was of a valid format, they were successful in uploading an ASPX web shell. When this newly uploaded web shell was accessed, it granted access to the underlying file system of the web server's operating system. Schmidt and Bosco then used the identified file upload vulnerability and uploaded a binary payload to the web server, again by editing the POST request to appear as if it was a legitimate file. Not detected by any anti-virus, this payload was then executed via the web shell to establish a reverse TCP connection from the web server to Rapid7's attacking server, thus granting full shell access to this server and access to the internal environment.

By chaining three application vulnerabilities together and taking advantage of insufficient operating system and network access controls, Rapid7 had demonstrated to the client their web application's ultimately critical weaknesses by gaining unfettered, internal network access from a remote location.

As a result of this penetration testing process, this client now has a clear understanding of and insight into the risks and implications they pose to the overall security posture of their entire organization. By stepping through this chained exploit process, clients can understand how a real-world attack might play out and how to remediate the issues and secure their customer's sensitive healthcare data.

MISCONFIGURATIONS

Misconfigured network services were almost exactly as common a finding in the surveyed engagements as software vulnerabilities, with just over two-thirds of client sites suffering from at least one common misconfiguration.



These issues represent a mix of configuration errors (with service misconfigurations and default account access at the top of the list) to a lack of adherence to modern network best practices, such as network segmentation, failing to enforce least privilege principles, and a lack of password and patch management.

Perhaps the most interesting reading of this finding is between the lines, when we sort the data by internal and external discoveries.

86 External Engagements	47 revealed at least one misconfiguration	54.7% misconfiguration rate
27 Internal Engagements	26 revealed at least one misconfiguration	96.3% misconfiguration rate
11 Mixed Engagements	9 revealed at least one misconfiguration	82.8% misconfiguration rate

Unlike the identified vulnerabilities of the previous section, most of these misconfigurations read like traditionally "internal-only" service misconfigurations with no clear distinction between "usually external" and "usually internal" issues. However, these misconfigurations were discovered at nearly exactly the same rate as the vulnerabilities were discovered between the internal and external engagements. One or two of these externally discovered misconfigurations is oftentimes enough to traverse the (largely imaginary) boundary from "outside" to "inside," and these issues should absolutely not be considered insider-only risks.

Learn more at:

www.rapid7.com/info/under-the-hoodie

UNDER THE HOODIE:
SMB Relay for Domain Admin and SSNs

Client Vertical: Manufacturing

While internal penetration testing can reveal critical software vulnerabilities present in the internal network, Rapid7's Bill Harshbarger, David Green, Patrick Kiley, and Austin Lane proved that it can also reveal critical network misconfigurations as well. We all want our personal information to be safeguarded, and we trust our employers to protect this information on our behalf. Harshbarger, Green, Kiley and Lane showed one client just how possible it is to abuse trust relationships to obtain access to employee PII.

The team began this engagement by looking for possible entry points that could give them a foothold on the network. To accomplish this, the team performed poisoning attacks on legacy Microsoft broadcast traffic such as Link-Local Multicast Name Resolution (LLMNR) and NetBios Name Service (NBNS). This allowed the team to intercept SMB challenge/response traffic from normal network traffic, and obtain 14 valid NetNTLMv2 password hashes for various users on the network. From here, these hashes were then sent to an offline password cracker for a dictionary-based attack. The team cracked six of the 14 weakest hashes, as those passwords were based on dictionary words.

With six valid accounts, Rapid7 began to iteratively determine where on the network these users had access, and whether any had administrative access. By cross-referencing usernames against the group memberships, Rapid7 determined which areas these users may have access to and began authenticating to servers throughout the environment utilizing these harvested credentials. Once on compromised machines, Rapid7 began scraping the memory of these servers and extracting plaintext credentials for other users that had logged in locally recently. This process continued until Rapid7 ultimately extracted a service account credential that enjoyed Domain Administrator privileges on the network. The team then extracted the NTDS.dit file from the Domain Controller, which contains the password hashes for every user on the network. These were then sent to the same offline password cracker for processing, which successfully cracked approximately two-thirds of the hashes. Armed with more user accounts and passwords, Rapid7 gained access to the client's entire network infrastructure, including the Radius authentication server and every router and switch in the environment.

The team used these maximally privileged accounts to identify business risks by looking for unprotected sensitive information on network shares, hosts, and available databases. The cracked hashes from the Domain Controller provided excellent search terms, since searching for plaintext passwords revealed many network-accessible, cleartext sources of authentication data.

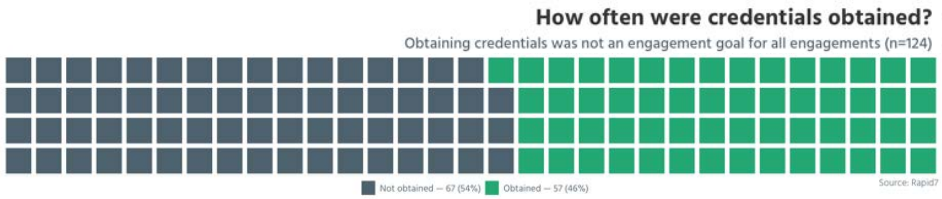
In addition, the team discovered sensitive documents on an internal server that contained customer credit card applications, many of which had full Social Security Numbers in cleartext, as well as wire transfer and banking information for both employees and customers of the client.

As a result of penetration testing, this client now has a clear understanding of the risks identified in their network environment and the severity of using weak passwords for user accounts. The client now also sees the damage service accounts can pose to the overall organization by being placed in the Domain Admin group, as they should have their own permissions and groups that provide them the least level of privilege required for normal functionality. With this information, the client is now better able to protect their employees and clients.

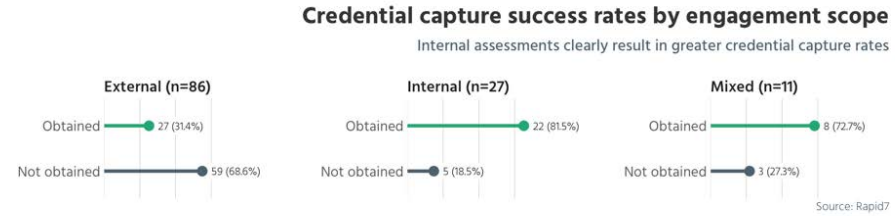
CREDENTIALS

Many penetration testers will tell you that vulnerabilities, misconfigurations, and exploits are not required to unearth and exfiltrate sensitive data from a target organization. Indeed, the most famous single breach in 2016 was the successful spear phishing campaign against Democratic campaign manager John Podesta, which leveraged nothing more than a look-alike password reset email[10]. No fancy exploits or hacking skills were required.

The loss of control over credentials remains the easiest for attackers to execute, and nearly half of the engagements (46.0%) resulted in compromised credentials. While this is lower than either leveraging network misconfigurations (67.2%) or exploiting vulnerable software (68.0%), compromised credentials are often either the object of compromising network services in the first place, or conducted via largely non-technical phishing or social engineering campaigns.

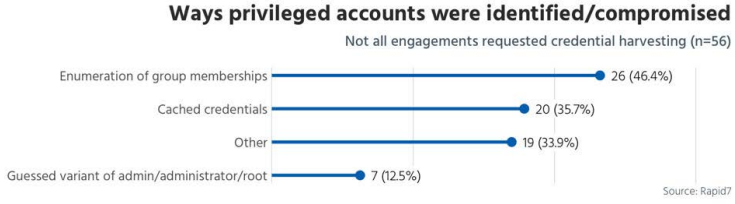


Internal assessments clearly result in greater credential capture rates, to the tune of 81.5% success versus an external assessment capture rate of 31.4%, as illustrated below. However, the fact that credentials can be compromised by external attackers nearly a third of the time is still alarming, given that there is no equivalent to a patch or a firewall rule for social engineering or directed phishing. Not only are these techniques effective, but they can be frustratingly difficult to address without some kind of incident detection and response solution in place.



Methods of Compromise

Credentials are generally two-part tokens[11]. First, the attacker needs to learn valid usernames and the naming convention in a given domain, then they associate valid passwords with those usernames.

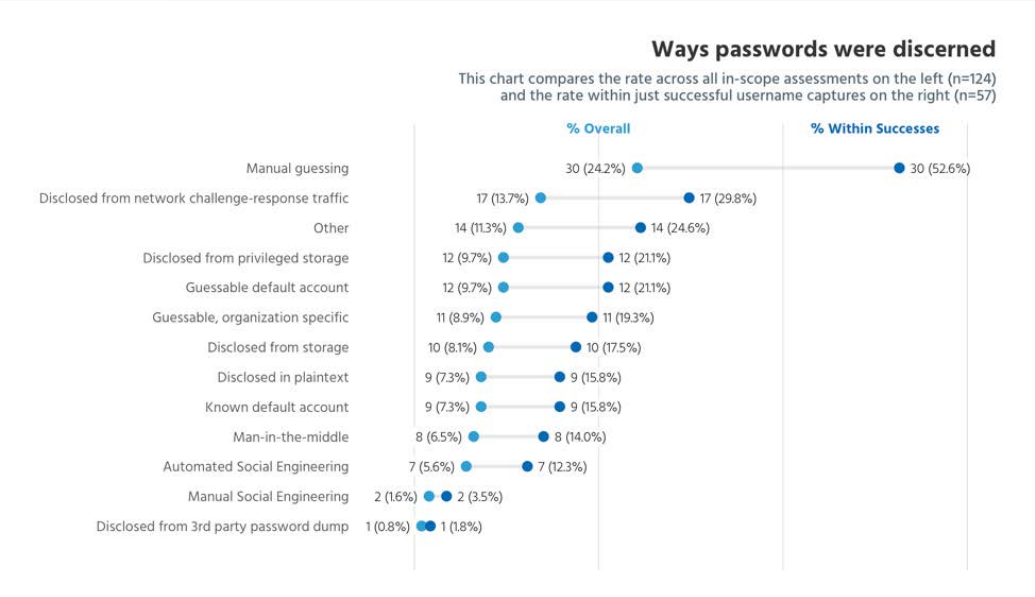


[10] http://motherboard.vice.com/read/how-hackers-broke-into-john-podesta-and-colin-powells-gmail-accounts

[11] Technically, the realm, or domain, to which a credential belongs is also part of the credential, but in virtually all engagements, the penetration tester is already aware of what organization he's attacking and can figure out domains rather quickly.

In the past, enumerating usernames was somewhat of a chore that required a bunch of noisy network-based reconnaissance. While this technique is still in use, it's becoming much more common to rely on some basic open source intelligence (OSINT) along with some educated guessing. Not only are these external sites reliable sources of username data, but leveraging them can avoid tripping any on-site intrusion detection systems (IDSes)[12].

Once valid username formats and usernames are learned, it's time to get to passwords. Here's a time-saving tip: If you know a lot of, or all, usernames, just try <Current season><current year>. People love that password, and according to our survey data, manually guessing patterns like this is successful a surprising (depressing?) fraction of the time.

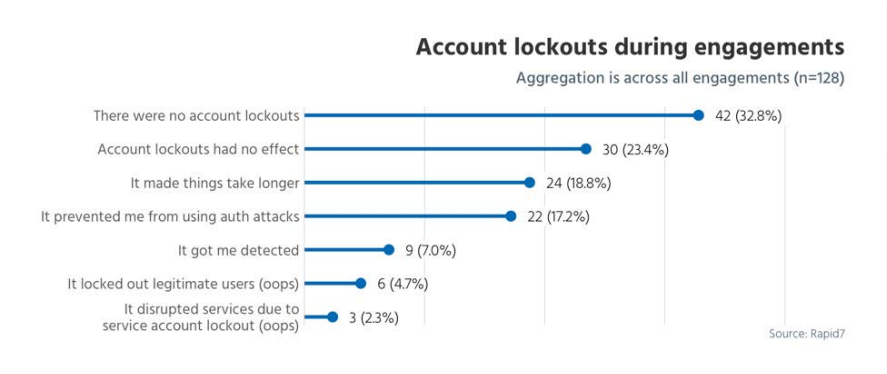


As we can see from the chart above, forms of human guessing using generic patterns, site-specific patterns, or common defaults net good credentials much of the time[13].

Enterprises do have some typical defenses against this sort of credential theft, chief among them being account lockouts and two-factor authentication.

Account Lockouts

The most common method for limiting credential guessing is lockout thresholds, which are supported by every desktop operating system and many of the most popular cloud-based services. Yet, as we can see from the chart below, many organizations either don't employ account lockouts at all (32.8% of the time), or the lockouts were not effective in limiting the penetration tester (23.4% of the time).



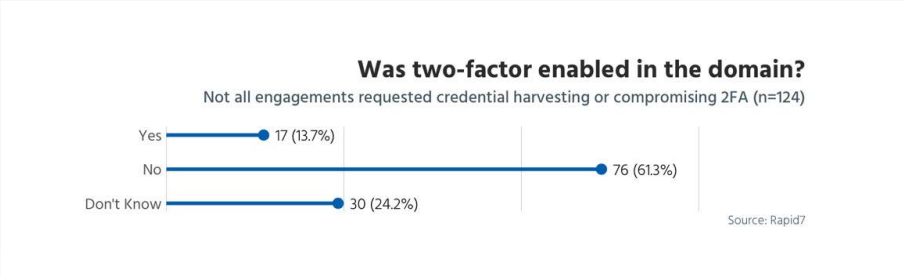
[12] Since multiple techniques are exercised, these percentages will go above 100% on the successful side.

[13] Again, multiple techniques means these figures will top out over 100% for successful compromises.consequences related to accidentally knocking life-saving medical gear offline, are almost certain to contribute to favoring assessing PII protections over operational assets.

While penetration testers were sometimes inconvenienced or detected thanks to lockouts, these security measures did end up disrupting normal services in at least a few cases. IT administrators are aware of this risk, of course, which may explain the surprisingly high lack of adoption. Also consider the misconfiguration findings earlier in this paper, where 14% of our surveyed sites lacked detection controls. Combined with a lack of effective lockouts, this is a prescription for inevitable compromise.

Two-Factor Authentication

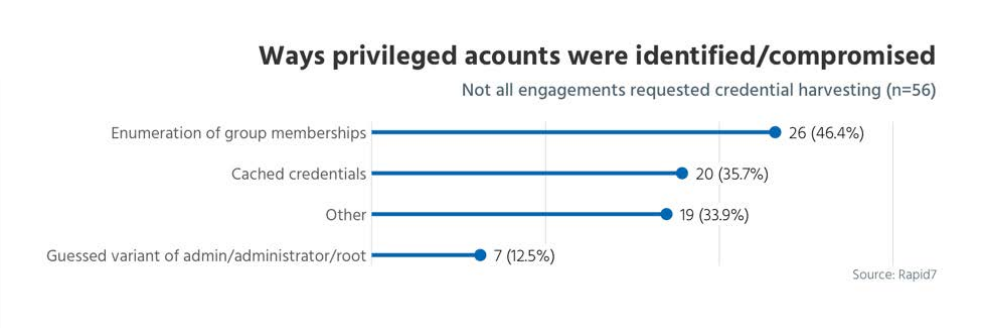
Safer than account lockouts, two-factor authentication (2FA) can dramatically enhance credential security when deployed. However, this is still an uncommon control to run into on an engagement.



Of the 85.5% of the engagements where credential theft was in scope, 2FA was simply not a factor. Considering the millions upon millions of large-scale breaches in 2016, and the endemic problem of password reuse, this finding was particularly disheartening. This is not to imply that 2FA is a magic bullet—service accounts, in particular, are often exempt from 2FA, and some isolated network assets or niche cloud services may not be compatible with the enterprise's chosen standard for 2FA. Even given these limitations, however, 2FA is generally effective in preventing the most common forms of credential compromise, especially when combined with a reasonable detection control like user behavior analytics.

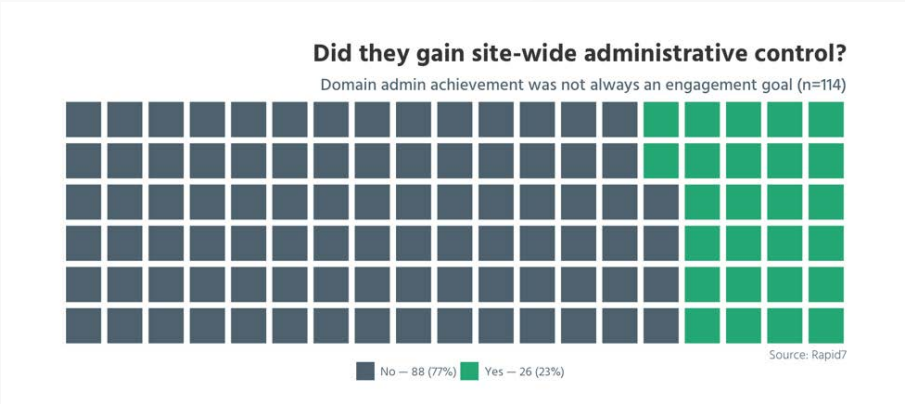
Escalating Privilege

Usually, penetration testers and attackers alike are not satisfied with the first account they are able to hijack; oftentimes, they will seek out more privileged accounts.



In the cases where privilege was obtained, merely guessing was not usually effective. These findings illustrate what we expect to find in a typical engagement: a lower-privilege account password is guessed, using "easy" methods, and then more sophisticated techniques are leveraged to gain elevated rights.

While locally privileged accounts are often sufficient to complete an engagement, domain administrator is usually the signal that the penetration tester has won an engagement thoroughly and completely. In nearly a quarter of engagements, this achievement was unlocked.



Taken with the finding that some kind of credential was acquired in 46% of engagements, this finding illustrates that once one credential is obtained, that credential can be parlayed into full domain administrator access about 45% of the time.

UNDER THE HOODIE:
Sidestepping 2FA and a Group Policy Preference Surprise

Client Vertical: Technology

External attack surfaces of technology companies usually have heightened information security standards and, ideally, their users are more savvy when it comes to choosing passwords. However, Rapid7's Nate Power and Patrick Laverty proved this was not the case for one particular organization. The team proved how good information gathering, coupled with precise password sprays, can ultimately result in going from an unauthenticated nobody on the internet, to an authenticated user on the Domain, and ultimately to a Domain Administrator.

As part of the reconnaissance phase, the penetration testers searched the Internet for any information that could help build a list of potential employee names or usernames for use in brute-force password-guessing attacks. Analyzing the contents of some of the collected files revealed information such as Active Directory usernames stored within the metadata. Knowing this username format, Power and Laverty created a list of potential usernames to use for password guessing.

Once the list of potential usernames was created, the pair then targeted Outlook Web Access (OWA) and used the list of usernames in a timing-based attack in an attempt to enumerate valid accounts in the domain. This username enumeration technique produced several valid accounts in the domain, which were then re-ran through a brute-force attack against the OWA installation using that favorite password of pen testers, <CurrentSeason><CurrentYear>. This attack produced several valid credentials pairs.

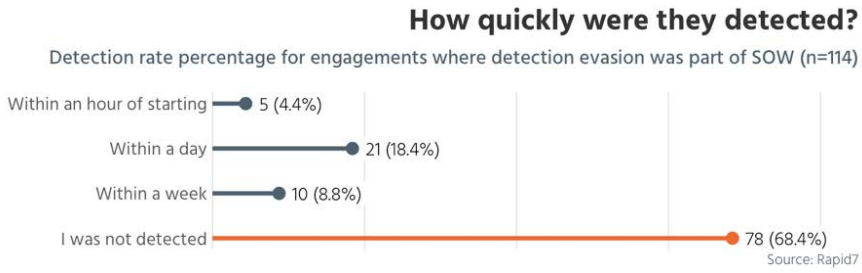
The client had a two-factor authentication (2FA) solution connected to its VPN endpoint, which would normally prevent logins with just a password. Unfortunately, the VPN also had a self-service enrollment portal that itself was not 2FA-controlled. Once this was discovered, it was a simple matter of changing a compromised account's associated email address to one controlled by the pen testers, verifying it, and then re-enrolling in 2FA, complete with a fresh OTP seed value.

Completing this process granted a credentialed foothold into the internal network. Once on the internal network, Power and Laverty scanned the internal hosts to ascertain the attack surface. They were successful in identifying the Domain Controller for the organization as well as several other servers. Upon checking these hosts for misconfigurations, a Group Policy Preference (GPP) file was discovered. This GPP was fairly old, predating MS14-025, and therefore contained service account credentials vulnerable to trivial decryption. This user was a Domain Administrator on the network, and therefore Rapid7 had fully compromised this domain upon connecting to the domain controller with this account.

This engagement proved to the client that their employee password discipline, as well as their 2FA deployment, needed some improvement. Rapid7 was able to provide guidance to mitigate these issues and increase the client's security posture to help prevent something like this from happening in the future.

THE STATE OF DETECTION

Most breaches are only detected weeks to months after the fact, if at all. This is well after the attacker has gotten a hold of whatever sensitive data he could find. This is one area where we believe penetration tests are significantly different from a more "real world" attack. The time box constraints described earlier in this paper tend to mean that penetration testers have to dial up their reconnaissance efforts to ludicrous speeds, taking risks that a disciplined, methodical attacker is unlikely to accept. As a result, about a third of engagements produced an alert that was noticed by the organization under test.



However, this is a rosy, glass-half-full reading of the census data. Even with time limits and a general disregard for stealth, resulting in a sudden tsunami of scan and exploit traffic, over two-thirds of penetration testers remain undetected. Beyond network segmentation, patch management, or any other technical countermeasure, a routine malicious behavior detection strategy that is at least able to catch these frenetic bursts of malicious activity is the best technical protection solution money can buy today.

Perhaps the most interesting detection finding that we saw in our data was that organization size didn't appear to affect detection rates at all; a smaller, nimbler organization with a correspondingly lower attack surface is no better at detecting penetration test activity than a large, well-resourced organization with a relatively large attack surface. Similarly, no particular industry seems better equipped to detect malicious activity than any other.

In either case, it's in the target organization's best interest to avoid the temptation to overtly tip off the security staff that penetration testing is about to occur. After all, it's not a penetration tester's job to shame or undercut the authority of the on-site IT security professionals. Instead, the penetration tester is there to help the regular IT security personnel prioritize the most effective countermeasures to the most likely attacks that network is going to face. In nearly all cases, this is best achieved by letting the penetration tester do the job of illuminating risk, vulnerability, and exposure, up to and including routine log analysis and other network detection strategies that are normally in place.

Watch the webcast to hear directly from the authors:

www.rapid7.com/info/under-the-hoodie

METHODOLOGY

The source material for this paper was drawn from post-engagement exit surveys conducted with the Rapid7 Global Services organization, from October, 2016 and the first week of January, 2017. We had 128 responding penetration testers. The fourth quarter is usually the busiest time of year for our penetration testers, and we felt that this timing would get us the most representative sample of our penetration testing clients (although not every engagement during the period is reflected in this census). We expect to continue to collect data from our penetration testers and build up our dataset over time in order to continue to better understand and share the realities of penetration testing. We also expect to adjust our exit survey questions as we learn how to ask better questions and collect more meaningful data from respondents.

As with any credible polling methodology, our questions were carefully crafted, with input from both data science and penetration testing experts. Our goals were to create a survey that would be easy for penetration testers to answer quickly as well as to capture results in a way that's useful for long-term statistical analysis.

Respondents filled out this survey immediately after primary penetration activities were complete. In most cases, surveys were taken within one to two days of completion of the engagement so that we could capture results while details were still fresh in the minds of the respective penetration testers.

The survey questions themselves are replicated in Appendix A, and as we explore cataloguing and quantifying penetration testing activities, we expect to eliminate and replace many of the free-form, difficult to analyze responses, and enhance the fidelity and accuracy of the remaining single and multiple choice questions.

ABOUT RAPID7

With Rapid7, technology professionals gain the clarity, command, and confidence to safely drive innovation and protect against risk. We make it simple to collect operational data across systems, eliminating blind spots and unlocking the information required to securely develop, operate, and manage today's sophisticated applications and services. Our analytics and science transform your data into key insights so you can quickly predict, deter, detect, and remediate attacks and obstacles to productivity. Armed with Rapid7, technology professionals finally gain the insights needed to safely move their business forward. To learn more about Rapid7, visit www.rapid7.com.

APPENDIX A: PENETRATION TESTING EXIT SURVEY

The survey conducted was 27 questions total, covering four topics. The survey is concerned with collecting anonymized, demographic data about the client, the use of credentials (and defensive measures such as two-factor authentication and account lockouts), the presence of vulnerabilities (and the exploits used to leverage them), and the abuse of misconfigurations. For ease of use, many questions depended on the positive answers to previous questions. For example, if the respondent did not leverage credentials in the engagement, none of the credential-specific questions would be asked.

Of special note is the first question, which collects the engagement ID number. From this, we're able to determine demographics about the client which are not asked directly; namely, the size, location, and business vertical of the organization under test.

1. What was the engagement ID?

Answer: Integer

2. How often is the client penetration tested?

Answer: Integer

3. How many penetration tests has the client had?

Answer: Integer

4. **Was the penetration test internal or external?**

Answer: Single choice

5. **How long was the engagement?**

Answer: Single choice

6. **What was the client interested in protecting?**

Answer: Multiple choice

7. **Were you able to obtain credentials?**

Answer: Single Choice

8. **How did you gather usernames?**

Answer: Multiple choice

9. **How did you obtain passwords?**

Answer: Multiple choice

10. **Did you compromise privileged accounts?**

Answer: Single choice

11. **How did you find privileged accounts?**

Answer: Multiple choice

12. **Was two-factor enabled in the domain?**

Answer: Single choice

13. **Were you able to bypass or compromise 2FA?**

Answer: Single choice

14. **How did you compromise 2FA?**

Answer: Single choice

15. **How effective were account lockouts?**

Answer: Multiple choice

16. **What kinds of vulnerabilities did you encounter?**

Answer: Multiple choice

17. **Did you encounter vulns not listed above? If so, please elaborate.**

Answer: Free-form text

18. **What misconfigurations did you leverage?**

Answer: Multiple choice

19. **If you leveraged misconfigurations not listed above, please elaborate.**

Answer: Free-form text

20. **What kinds of exploits did you use?**

Answer: Multiple choice

21. **If you used exploits not covered by the above, please elaborate.**

Answer: Free-form text

22. **Were you able to collect any confidential data?**

Answer: Single choice

23. **What kinds of data did you acquire?**

Answer: Multiple choice

24. **Did you collect any other data not described above? If so, please elaborate**

Answer: Free-form text

25. **Did you gain site-wide admin/root such as domain admin?**

Answer: Single Choice

26. **If so, very briefly, how did you gain site-wide admin/root?**

Answer: Free-form text

27. **How quickly were you detected?**

Answer: Multiple choice

LEARN MORE

at www.rapid7.com/info/under-the-hoodie