Legaler Einbruch – Pentesting

Schwarze Box

Legaler Einbruch: So kann ein Pentest aussehen

Schwarze Box

Unternehmen bezahlen Pentester dafür, dass sie versuchen, in ihre Systeme einzubrechen. So dubios das für Außenstehende klingen mag – Penetrationstests sind seit Jahren ein etabliertes Mittel, die Sicherheit von Systemen und Netzwerken zu überprüfen. Im Folgenden erfahren Sie, wie ein sogenannter Bl…

Unternehmen bezahlen Pentester dafür, dass sie versuchen, in ihre Systeme einzubrechen. So dubios das für Außenstehende klingen mag – Penetrationstests sind seit Jahren ein etabliertes Mittel, die Sicherheit von Systemen und Netzwerken

zu überprüfen. Im Folgenden erfahren Sie, wie ein sogenannter Black-Box-Test abläuft.

Von Michael Wiesner

kompakt

- Pentester sind vom Betreiber eines Netzwerks beauftragte Hacker.
- Sie nutzen dieselben Werkzeuge wie echte Angreifer und decken Schwachstellen auf.
- Lediglich ausgestattet mit der Domain will Michael Wiesner in die internen Systeme seines Auftraggebers einbrechen.

Es ist April 2022, ich will eigentlich gerade den Laptop zuklappen, da flattert eine Anfrage in mein Postfach. Ein mittelständisches Maschinenbauunternehmen aus dem Norden Deutschlands will mich für einen sogenannten Pentest buchen.

Ein Pentest kann vieles sein, das Spektrum reicht von Sicherheitsanalysen einzelner Applikationen oder Systeme bis hin zur Simulation zielgerichteter Angriffe. Noch umfassender sind sogenannte „Red Team Assessments“. Dabei überprüfen Pentester, wie gut Systeme und Mitarbeiter zur Erkennung und Abwehr von Angriffsversuchen ausgerüstet sind.

Im Videotelefonat am nächsten Tag schildert der Chef der IT-Abteilung des Auftraggebers, worum es geht: Ich soll ohne Kenntnis über die IT-Infrastruktur in interne Systeme des Unternehmens einbrechen. Als einziger Anhaltspunkt dient die Domain – eine Information, die jeder Mensch mit Zugang zum Internet innerhalb von Sekunden herausfinden könnte. „Black-Box-Test“ nennt man solche Penetrationstests, bei denen der Pentester agiert wie ein typischer Angreifer. Alle weiteren benötigten Informationen muss ich dabei – in Abgrenzung zum White-Box-Test, bei dem der Pentester über Insiderwissen verfügt – selbst herausfinden. Der Einbruchsversuch soll einen zielgerichteten Angriff simulieren und möglichst verdeckt über

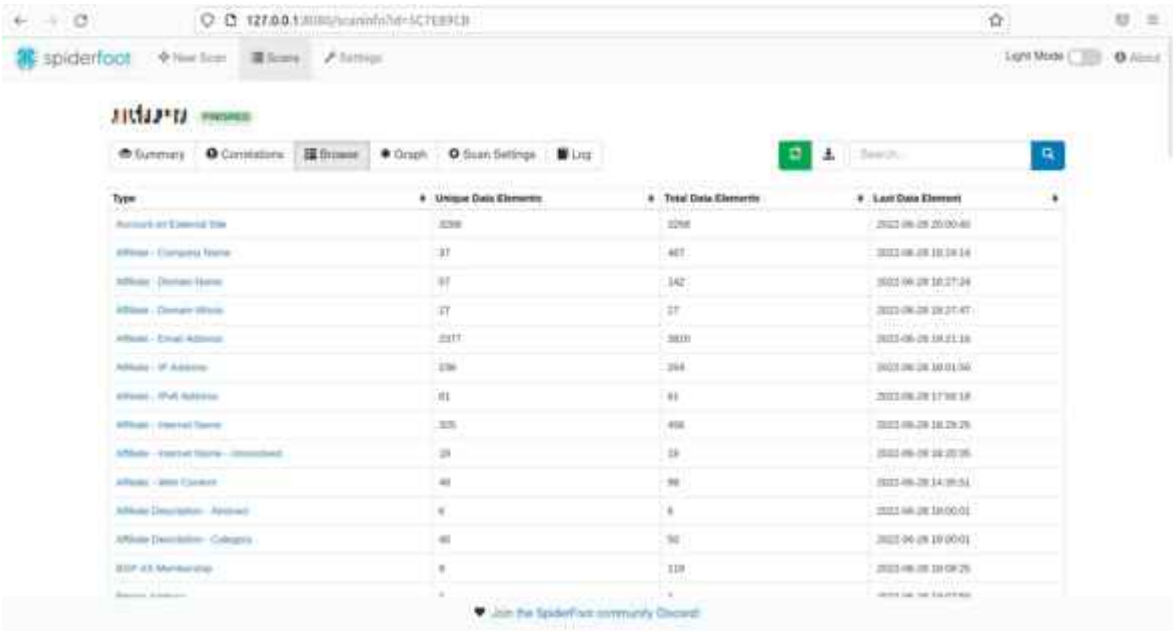
das Internet erfolgen. Entsprechend darf ich alle Mittel einsetzen, die auch ein echter Angreifer nutzen würde. Das könnte spannend werden – ich bin interessiert. Wir besprechen die Rahmenbedingungen und halten das Ganze vertraglich fest. In Angriff nehme ich den Test Anfang Juni.

Bei der Simulation eines solchen zielgerichteten Angriffs orientiere ich mich an den Phasen der MITRE ATT&CK Matrix. Darin werden die Taktiken und Techniken echter Cyberangriffe beschrieben und kategorisiert. Sie dient als Wissensdatenbank, die Verteidigern dabei hilft, Bedrohungen zu erkennen und abzuwehren, leistet mir bei einem Pentest, bei dem ich schließlich selbst in der Angreiferrolle stecke, aber ebenso gute Dienste.

Informationsbeschaffung

Ich starte mein Hacking-Vorhaben, indem ich alle frei verfügbaren Quellen nach Informationen über mein Ziel durchkämme. Dafür gibt es im Netz eine Reihe von Websites, Diensten und Datenbanken. „Open Source Intelligence“ (OSINT) nennt man diese Art der Informationsgewinnung auch (siehe c’t 16/2022, S. 138). Abfragen an WHOIS-Datenbanken und DNS-Server liefern mir erste Anhaltspunkte, mithilfe der Tools dnsrecon, spiderfoot und Shodan automatisiere ich einen Großteil der Arbeit. Das Python-Skript dnsrecon füttere ich im Bruteforce-Modus mit der Domain und einer Wortliste – es fragt Subdomains und Hostnamen ab und wertet praktischerweise anschließend gleich aus, welche IP-Adressen sich dahinter verbergen. Wie erwartet, liefert das Skript, und ich erhalte eine recht umfassende Liste der öffentlich erreichbaren Systeme meines Auftraggebers. Über die Kommandozeile rufe ich das OSINT-Tool Spiderfoot auf. Es nutzt eine größere Anzahl von Quellen für die Informationsgewinnung. Zum Beispiel ermittelt es mögliche Hostnamen auch durch die verwendeten TLS-Zertifikate. Auch liefert das Tool gültige Mailadressen, Telefonnummern, ähnliche oder verbundene IP-Adressen und Domains (siehe Bild

unten). Auch der Webdienst Shodan – sicher das prominenteste Beispiel für solche Scanner – liefert umfangreiche Informationen über die öffentlich erreichbaren Systeme meines Auftraggebers, die ich möglicherweise für den eigentlichen Angriff nutzen kann.



Spiderfoot ist ein OSINT-Tool, das dem Nutzer gleich eine ganze Palette an Informationen liefert, darunter auch Mailadressen.
Meine Zugriffe auf die öffentlichen Webseiten meines Auftraggebers werden zwar sehr sicher protokolliert, jedoch nicht blockiert, also kann ich davon ausgehen, dass sie nicht als schädlich erkannt werden. Zahlreiche Unternehmen, Organisationen und Einzelpersonen durchforsten das Internet laufend nach interessanten Systemen, Anwendungen oder Inhalten, sodass ein gewisses Grundrauschen besteht. Meine vorsichtig durchgeführten Verbindungsversuche gehen offenbar darin unter.

Vorsichtig anklopfen

Die gesammelten Daten verraten mir bereits eine ganze Menge über die Systeme und Anwendungen, die mein Auftraggeber verwendet, denn aus den DNS-Hostnamen kann ich ableiten, um welche Dienste es sich handelt: Bei „owa“ kann ich davon ausgehen, dass ein Exchange-Server betrieben wird und dieser

über „Outlook Web Access“ im Internet zur Verfügung gestellt wird. Generische Namen, wie „vpn“ oder „sslvpn“ sind selbsterklärend, „citrix“ weist auf ein Remote-Access-Gateway des gleichnamigen Herstellers hin. Teilweise sind die Hostnamen gleich, aber durchnummeriert – etwa owa2. Das könnte ein Hinweis darauf sein, dass mein Auftraggeber mehrere Versionen einer Anwendung einsetzt, oder darauf, dass ein Dienst über unterschiedliche Internetanbindungen bereitgestellt wird.

Die so gewonnenen Informationen über die Systeme und Dienste meines Auftraggebers sind noch nicht komplett. Aber um erste Angriffe zu starten, reichen sie aus. Für eine vollständigere Übersicht fehlt mir die Zeit – für den Penetrationstest sind lediglich sechs volle Arbeitstage veranschlagt – etwa zwei davon habe ich bereits für die Reconaissance-Phase, wie man die Phase der Informationsgewinnung im Fachjargon nennt – bereits aufgebracht. Für eine möglichst vollständige Übersicht muss man Ports scannen, beispielsweise mittels nmap oder einem Schwachstellenscanner wie Nessus oder OpenVAS. Die Herausforderung dabei ist, diese Scans so vorsichtig wie möglich durchzuführen, um weiterhin unentdeckt zu bleiben. Konkret bedeutet das, dass der Scan über einen langen Zeitraum verteilt werden muss, weil nur wenige gleichzeitige Verbindungen aufgebaut werden dürfen.

Da der vereinbarte Umfang und Zeithorizont des Penetrationstests kein solches Vorgehen erlaubt, wende ich die Holzhammermethode an: Über einen nur für diesen Vorgang genutzten Internetzugang starte ich ohne Rücksicht auf Verluste einen Portscan auf alle IP-Adressen und Ports. Dieser bleibt wegen des schon erwähnten Grundrauschens tatsächlich unbemerkt, liefert aber leider nicht die gewünschten Ergebnisse. Die „Portscan Protection“ der Firewall meines Auftraggebers blockiert den Scan erfolgreich. Das merke ich daran, dass mein Scan anfänglich zwar Ergebnisse liefert, die Systeme nach einer gewissen Zeit aber aufhören zu antworten.

Die Portscan Protection verlangsamt meine Verbindungsanfragen entweder stark oder blockiert sie ganz.

Für den Einstieg in die nächste Phase, die der Schwachstellenidentifikation, bleiben mir also nur die bereits gewonnen Informationen. „Schade“, denke ich – ganz so leicht scheint der norddeutsche Mittelständler es mir an dieser Stelle nicht zu machen.

Schwachstellensuche

Auch bei der Suche nach Schwachstellen greife ich auf Shodan zurück. Ich nutze den Webdienst, um auf Basis der Versionsnummern zu ermitteln, ob es bekannte Schwachstellen in den Diensten gibt. Leider ohne Treffer – laut Shodan hat nicht einer davon eine Schwachstelle, die ich für einen Angriff ausnutzen hätte können.

Ich muss wohl doch etwas genauer hinschauen: Mithilfe von Shodan, Spiderfoot und Dnsrecon habe ich knapp 70 offene Ports identifiziert. Das heißt, etwa 70 erreichbare Dienste warten darauf, genauer unter die Lupe genommen zu werden. Ich gehe systematisch vor. Zunächst filtere ich Dienste heraus, die nicht selbst betrieben werden und damit auch keinen potenziellen Zugriff auf die IT-Infrastruktur erlauben, wie etwa die Website beim Hoster. Die hebe ich mir für später auf, für den Fall, dass ich keinen direkten Weg in das Netzwerk meines Auftraggebers finden sollte. Die verbleibenden Portnummern geben Preis, um welche Art von Dienst es sich handelt. Um herauszufinden, welche Software und Version sich dahinter verbirgt, setze ich die Kommandozeilentools netcat oder alternativ telcat ein.

Zunächst surfe ich die identifizierten Webserver allerdings manuell über einen Webbrowser an, um zu erfahren, was sich hinter der URL tummelt. Dabei achte ich peinlich genau darauf, dass ich nicht nur die IP-Adresse, sondern auch den jeweilige Fully-Qualified Domain Name, kurz FQDN verwende. Dieser

vollständige Domainname einer Internetpräsenz ist eindeutig und er lässt sich den zum Nameserver gehörenden IPv4- oder IPv6-Adressen zuordnen. Das ist wichtig, weil oft mehrere unterschiedliche Webserver hinter der gleichen IP-Adresse betrieben werden – ohne die Angabe des FQDN würde ich möglicherweise nicht an die gewünschten Informationen kommen. Weil ich auf diese Weise – im Unterschied zum fehlgeschlagenen „Holzhammerscan" – nur noch einzelne Ports kontaktiere, könnte ich nun eigentlich umfangreiche Schwachstellenscans durchführen, ohne dass die Firewall wieder den Riegel vorschieben würde. Die Betonung liegt auf eigentlich – denn schon als ich die Websites manuell ansurfe, lande ich auf Login-Pages – ein deutlicher Hinweis darauf, dass mein Auftraggeber einen Reverse Proxy verwendet, um die Webseiten im Internet zu veröffentlichen. Das bedeutet, dass die Server nicht direkt angesprochen werden, sondern die Verbindungen vorab von einer Stellvertretersoftware angenommen werden. Schlimmer noch: Über die Eingabe bestimmter Parameter fingiere ich eine Directory-Traversal-Attacke und finde heraus, dass der Reverse Proxy zusätzlich über eine sogenannte Web Application Firewall verfügt. Das sind Systeme zur Erkennung und Abwehr von Angriffen, kurz WAF.

Das trübt meine Aussichten auf einen erfolgreichen Angriff über Sicherheitslücken in Web-Applikationen erheblich. Zähneknirschend verzichte ich auf einen umfangreichen Schwachstellenscan der Web-Apps – schließlich will ich die WAF nicht alarmieren. Ich verwende nikto und ein kommerzielles Tool namens Nessus, um die Websites auf Schwachstellen zu prüfen, werde jedoch nicht fündig. Komplett ertragslos verläuft meine Schwachstellensuche zum Glück trotzdem nicht. Beim manuellen Ansurfen haben Login-Pages mir verraten, dass es sich bei drei der Websites des Auftraggebers um Fernzugriffsportale handelt. Mithilfe von Nessus scanne ich sie ebenfalls auf Schwachstellen und gleiche zusätzlich die Versionsnummern mit der CVE-Datenbank https://cve.mitre.org ab. Leider fördert keine meiner Bemühungen eine

Sicherheitslücke in einem der Fernzugriffsportale zutage, aber ich nehme mir trotzdem vor, diese vorerst im Hinterkopf zu behalten.

Ein Schritt vor und zwei zurück

Die ernüchternde Zusammenfassung bis zu diesem Punkt: Keines der öffentlich erreichbaren Systeme des Auftraggebers besitzt offensichtliche Schwachstellen, die ich direkt zum Einbruch in die Systeme oder das Netzwerk hätte nutzen können. „Wäre ja auch zu leicht gewesen“, denke ich, während ich meine Optionen für das weitere Vorgehen abwäge. Ich habe nicht mehr viel Zeit, knapp zwei Drittel der maximal veranschlagten sechs Arbeitstage sind bereits verstrichen. Eine aufwendige Untersuchung der restlichen Webseiten, zum Beispiel mittels der beliebten Burp Suite fällt daher flach. Ein erneuter Blick auf die vereinbarte Leistungsbeschreibung zaubert mir dann aber doch ein Lächeln auf die Lippen. Fast hätte ich es vergessen, aber dort steht schwarz auf weiß, dass ich auch Phishingmethoden einsetzen darf. Phishing hat in der Regel das Ziel, den Empfänger dazu zu verleiten, Informationen preiszugeben oder ihn dazu zu bringen, Dateianhänge oder Links auf verseuchte Websites anzuklicken, über die dann Schadprogramme – sogenannte Remote-Access-Trojaner, kurz RAT – ausgeführt werden, die einen Zugang zum betroffenen System öffnen.

Spiderfoot hat mir während der Informationsbeschaffungsphase bereits eine Liste von circa 20 E-Mail-Adressen geliefert, denn auf der Webseite des Auftraggebers werden einige Mitarbeiter mitsamt der E-Mail-Adressen präsentiert. Das ist ein guter Start, aber ich würde die Angriffsfläche gerne vergrößern. Dabei spielen mir die beliebten Business Social Networks Xing und LinkedIn in die Hände. Anhand der Spiderfoot-Liste weiß ich ja bereits, wie die Unternehmens-Mailadressen aufgebaut sind und mit den Namen der Beschäftigten aus den Business-Netzwerken kann ich über ein

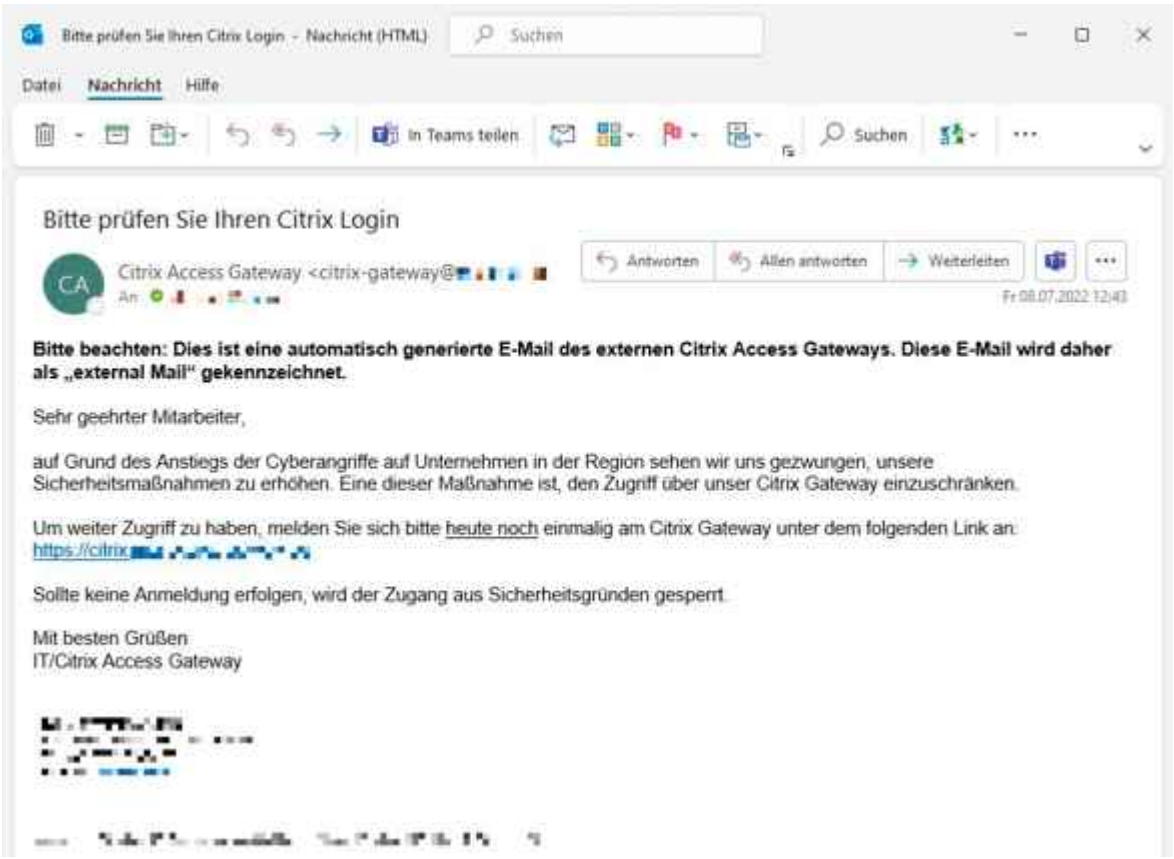
einfaches Skript leicht 50 weitere Mailadressen generieren.

Grundsätzlich sind solche Business-Portale ein Quell nützlicher Informationen. Über die eingetragenen Kenntnisse, Fähigkeiten oder die „ich biete“-Felder der Mitarbeiter lässt sich oft herausfinden, welche IT-Hersteller und Produkte eingesetzt werden. Dies ist besonders hilfreich, wenn die Hersteller von Firewalls, E-Mail-Gateways oder Endpoint-Security-Produkten genannt werden. Die Wahrscheinlichkeit ist hoch, dass diese dann auch in deren Unternehmen eingesetzt werden.

Eigene RATs zu erstellen, die nicht von der verwendeten Antivirussoftware erkannt werden, ist längst kein Hexenwerk mehr, so es sich denn um einen klassischen Virenschutz handelt. Ich versuche, den Trojaner über eine verschlüsselte Zip-Datei an der Firewall – beziehungsweise dem E-Mail-Security-Gateway – vorbeizuschmuggeln und erwarte fast, so mein Ziel zu erreichen, schließlich hat mich diese Methode in der Vergangenheit schon oft zum gewünschten Ziel geführt. Nicht jedoch bei diesem Penetrationstest. Mein Auftraggeber filtert sehr erfolgreiche alle E-Mails mit entsprechenden Dateianhängen heraus. Damit nicht genug – eine schnelle Recherche über den Webdienst mxToolbox zeigt, dass die absendende IP-Adresse bereits kurz nach den ersten Versuchen auf den bekannten Blacklisten landet. Leider führt auch das Einschleusen bösartiger Links für einen „Drive By Exploit“ nicht zum Ziel. Mir wird klar, dass ich es anders versuchen muss. Nur wie? – Die zuvor identifizierten drei Fernzugriffsportale kommen mir in den Sinn. Sie böten ideale Ansatzpunkte für eine Phishing-Kampagne, die auf das Abfischen von Zugangsdaten abzielt.

Ich mache mich an das Basteln einer weiteren Phishing-Mail. Sie soll so offiziell wie möglich aussehen – inklusive farblich passendem Layout und einer authentisch wirkenden Signatur. Von einer Wegwerf-Mailadresse frage ich höflich bei der jobs@-Mailadresse des Unternehmens an, an wen ich denn

meine Initiativbewerbung schicken könne. Wie erwartet, bekomme ich eine freundliche Antwort mit der gewünschten Information – und der Standardsignatur des Auftraggebers.



Mittels einer Phishing-Mail wird versucht, die Mitarbeiter des Auftraggebers auf eine Fake-Website zu locken.
Außerdem brauche ich eine Phishing-Webseite, in die mindestens eines meiner Opfer hoffentlich die Zugangsdaten eingeben wird. Gefälschte Webseiten bekannter Dienste, wie Microsoft Office 365, Facebook, Instagram oder Twitter lassen sich leicht über freie Tools wie zphisher erzeugen.

Zphisher erstellt Fake-Websites bekannter Dienste. Anpassen lassen sich diese aber leider nur bedingt.
In der Phishing-Mail müsste man dann nur noch auf die entsprechende Webadresse verweisen, um die eingegebenen Zugangsdaten anschließend abrufen zu können. Anpassungen, wie zum Beispiel das Logo meines Auftraggebers einzubinden, kann man daran aber leider nur bedingt vornehmen – für meine Zwecke kann ich zphisher deshalb leider nicht nutzen. Stattdessen erstelle ich manuell einen Klon von einem der verwendeten Fernzugriffsportale.

Eine geklonte Fake-Website soll die Phishing-Opfer dazu verleiten, ihre Zugangsdaten einzugeben.
Bleibt noch die Frage, auf welcher Adresse das gefakte Portal betrieben werden soll. Die Originaladresse lautet „citrix.DOMAIN.com“ – für meinen Klon verwende ich kurzerhand eine ähnlich aussehende Adresse: „citrix-DOMAIN.com“. Solche Doppelgänger-Domains sind auch bei realen Angreifern beliebt, da man auf den ersten Blick den Unterschied zwischen echten und gefälschten Adressen nicht erkennt.

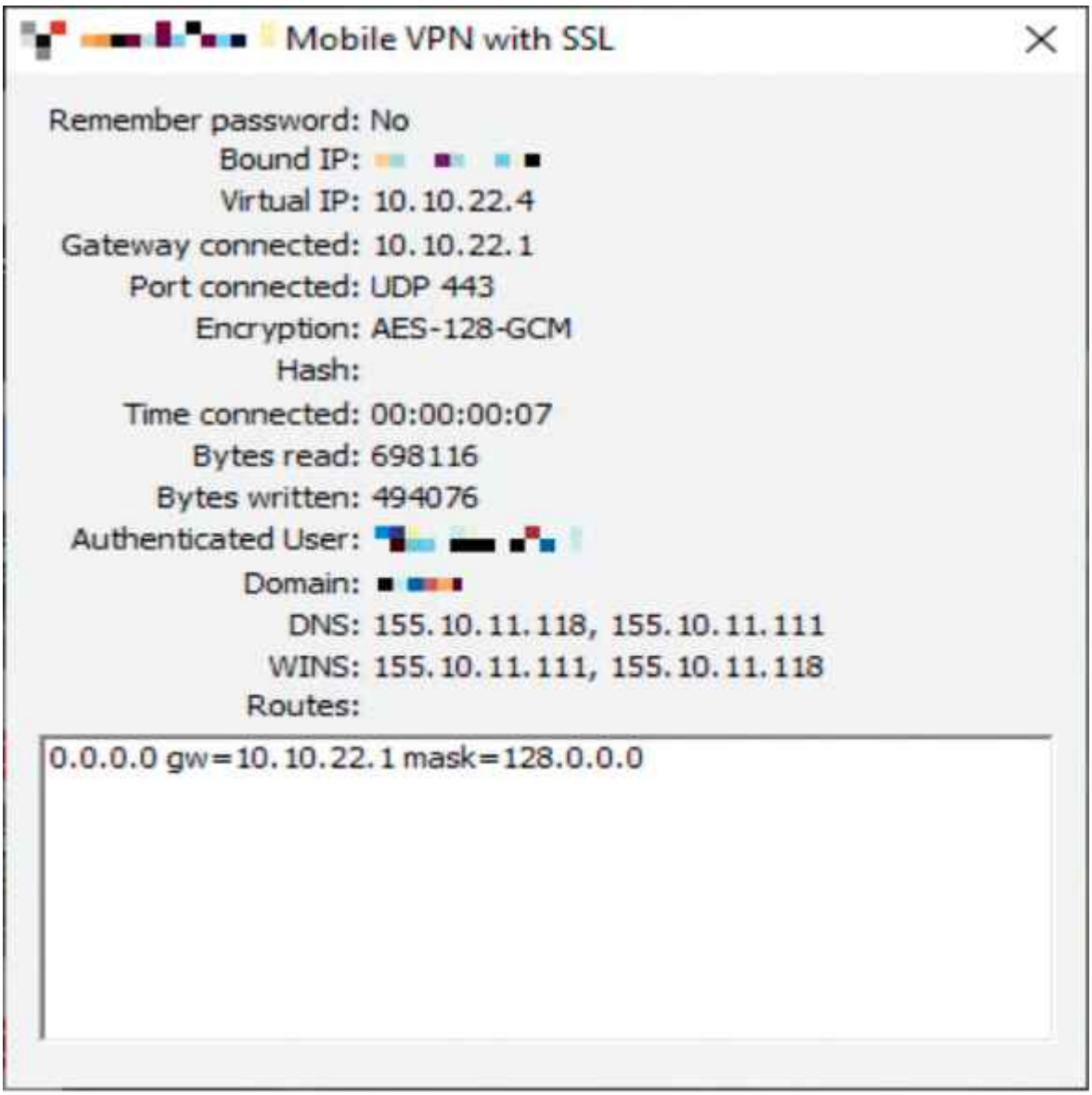
Bevor ich im großen Stil loslegen kann, gilt es vorab zu prüfen, ob die fingierte E-Mail die Empfänger überhaupt erreicht, oder ob sie – wie meine ersten beiden Versuche – durch Sicherheitssoftware blockiert wird. Ein Test mit ausgewählten Empfängern bringt die Ernüchterung: Das E-Mail-Gateway blockiert meine Phishing-Mails. Eine Überprüfung mittels mxToolbox verrät mir, dass meine IP erneut auf der Blacklist gelandet ist. Ich gerate ins Grübeln und ärgere mich zugegebenermaßen ein wenig, weil ich nicht gleich dahinter komme, warum. Die E-Mail war standardkonform, die IP-Adresse des Servers befand sich bis dato auf keiner Blacklist und auch der Text der E-Mail war nicht besonders spammy. Aber das E-Mail-Gateway ist anscheinend schlauer als gedacht: Ich vermute, es deklariert den eingebetteten Link auf die gefälschte Webseite als gefährlich, weil er Teile des Domainnamens des Auftraggebers enthält, jedoch nicht zu dessen

Servern gehört. Mir bleibt nur die Registrierung und Nutzung einer unverdächtigen Domain, die lediglich „citrix“ als Hostname aufführt und sonst möglichst offiziell aussieht. Ein Test mit dieser URL verläuft erfolgreich: Die E-Mails werden zugestellt. Anhand eintreffender Abwesenheitsnotizen kann ich zudem erkennen, dass die E-Mails nicht als Spam markiert wurden. „Feuer frei!“, denke ich grinsend.

Die eigentliche Phishing-Kampagne starte ich an einem Montag um 9 Uhr – pünktlich zum üblichen Arbeitsbeginn der Verwaltung. Und Bingo: Die ersten Zugangsdaten werden um 9:39 Uhr eingegeben. Jetzt darf ich keine Zeit verlieren. Wie klein mein Zeitfenster ist, kann ich nicht abschätzen, aber ich muss handeln, bevor es möglicherweise jemandem auffällt, dass Phishing-Mails im Umlauf sind und die Mitarbeiter – inklusive meiner Opfer – aufgefordert werden, ihre Zugangsdaten zu ändern.

Open the gates

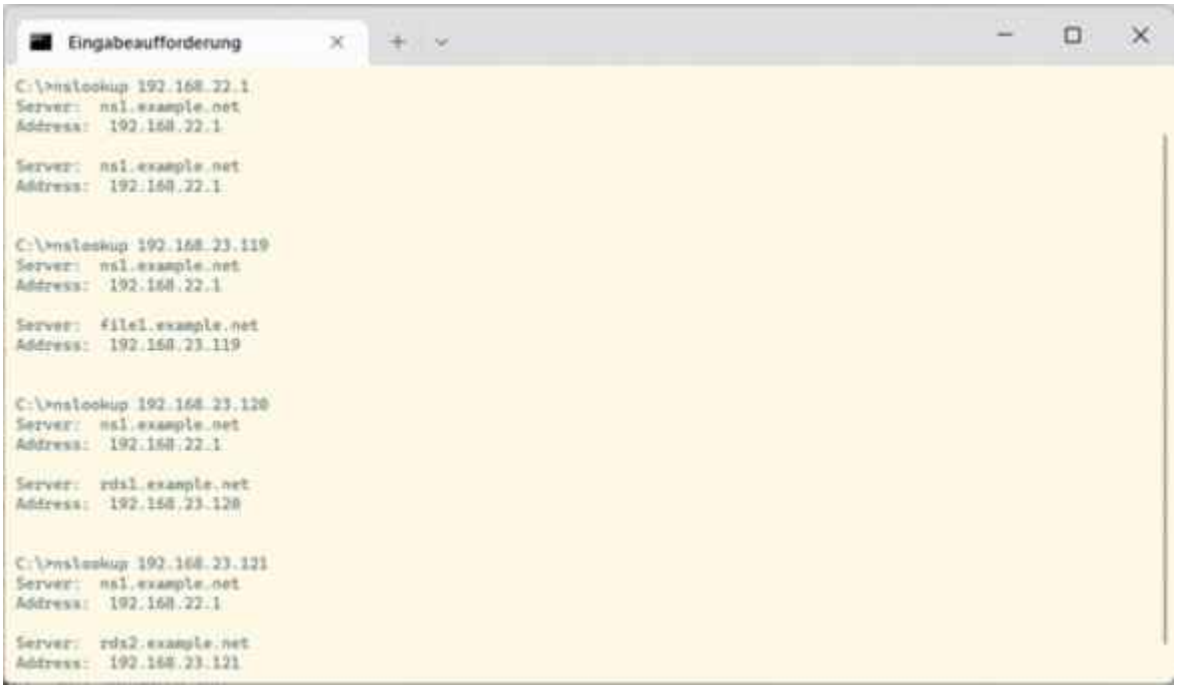
Meine Euphorie verfliegt, als die wirkliche Citrix-Anmeldeseite nach Eingabe der erbeuteten Zugangsdaten „Bitte Einmalpasswort eingeben“ meldet. Das Portal ist per Multi-Faktor-Authentifizierung abgesichert – und den zur Anmeldung benötigten zweiten Faktor besitzt nur der legitime Nutzer. Eins zeigt die Meldung jedoch: Die Zugangsdaten stimmen. Also versuche ich mein Glück beim nächsten Fernzugangsportal. Meine Hoffnung schwindet, als es ein Einmalpasswort anfordert. Aber ein Portal bleibt mir noch. Ohne große Hoffnung gebe ich die erbeuteten Zugangsdaten auch hier ein und halte die Luft an. Aber ich habe Glück: Nach dem Klick auf Enter zeigt mir das Portal einen Download-Link zum VPN-Dienst, den mein Auftraggeber verwendet. Ich installiere die Software, gebe die erbeuteten Zugangsdaten auch an dieser Stelle ein – und Bingo! – Der VPN-Client baut eine erfolgreiche Verbindung zum internen Netzwerk des Auftraggebers auf. Ich atme tief durch. Die nächsten Schritte muss ich sorgfältig planen.



Bingo! Das VPN-Gateway baut eine Verbindung zum internen Netzwerk auf.

Es gilt jetzt, das interne Netzwerk zu erkunden, um interessante Systeme und Daten zu identifizieren. Die Gefahr, erkannt zu werden, steigt dabei mit der Aggressivität des Vorgehens und den verwendeten Tools. Werden in dieser Phase Schwachstellenscanner oder Angriffswerkezuge, wie zum Beispiel Metasploit eingesetzt, ist die Gefahr groß, dass Endpoint-Security-Systeme oder vorhandene Intrusion-Detection-Systeme im Netzwerk dies erkennen, melden und anschließend die Verbindung zum internen Netzwerk getrennt wird. Unfehlbar sind solche Sicherheitsvorrichtungen allerdings nicht. Es ist möglich, sie auszutricksen, indem man sich als Angreifer verhält wie der eigentliche Anwender.

Aber ich will zuerst die wichtigste Frage klären – und dafür brauche ich sowieso noch keine Tools: Handelt es sich bei den VPN-Zugangsdaten auch um die Windows-Zugangsdaten? Klarheit verschafft mir die Anmeldung am Netlogon-Verzeichnis des Domain-Controllers. Es enthält in der Regel die Anmeldeskripte und kann daher auch von jedem Domain-Benutzer gelesen werden. Die Anmeldung funktioniert – ich habe tatsächlich die Zugangsdaten der Windows-Domain erbeutet.



Das Tool nslookup findet nach Eindringen in das Netzwerk die Namen weiterer Server heraus.

Jetzt könnte ich über PowerShell-Skripte oder Tools wie BloodHound oder PingCastle das Netzwerk und die Windows-Domain nach Schwachstellen durchforsten (siehe ct.de/yhxe). Ich entscheide mich aber lieber für die vorsichtigere Variante und sehe mich erst einmal manuell um. Im gleichen Netzbereich wie die Domain-Controller befinden sich üblicherweise auch weitere Server, deren Namen man durch einen Reverse-Lookup mittels des Standard-Tools nslookup auflösen kann. Durch den Aufbau der Hostnamen kann ich Rückschlüsse auf weitere Server ziehen. „rds1“ verweist etwa auf den ersten Terminalserver (Remote Desktop Services); „file1“ auf den ersten Fileserver (vergleiche Bild auf Seite 116). Getreu dem Motto „wer nicht wagt, der nicht gewinnt“, versuche ich mich auf den ersten

Terminalserver einzuloggen – und habe Erfolg. Der Server präsentiert die Standard-Arbeitsumgebung des unglückseligen Benutzers, dessen Zugangsdaten ich abgefischt habe – mitsamt allen Applikationen. Ich habe kompletten Zugriff im Kontext des ausgespähten Benutzers, inklusive E-Mail, Dateiablage, ERP-Software und Microsoft-365-Diensten, wie Sharepoint Online und Teams. Zudem offenbart ein kleines blaues Doppelpfeil-Symbol in der System Tray, dass der Auftraggeber das Support-Werkzeug TeamViewer einsetzt. Es könnte mir als mögliche Hintertür dienen, falls der Angriff erkannt und das VPN-Gateway abgeschaltet wird. Ein Blick in den Task-Manager des Servers zeigt die laufenden Dienste einer marktführenden „Endpoint Detection and Response"-Software, kurz EDR. Mein vorsichtiges Vorgehen war also mehr als angebracht. Der Versuch, zwischengespeicherte Authentifizierungsdaten auszulesen, zum Beispiel mit dem beliebten Tool Mimikatz, hätte höchstwahrscheinlich dazu geführt, dass ich entdeckt und aus dem System ausgesperrt worden wäre.

Bei einem klassischen „Double Extortion"-Angriff von Cyberkriminellen würden diese nun beginnen, die gefundenen Dateien zu exfiltrieren, sich im Netzwerk weitere Berechtigungen zu verschaffen, möglicherweise die Datensicherung zu manipulieren und Daten zu verschlüsseln, um anschließend Löse- beziehungsweise Schweigegeld zu fordern. Die Exfiltration von Daten bleibt meistens unerkannt. Ich simuliere den Vorgang, indem ich mehrere Gigabyte Daten in ein Zip-Archiv packe und es anschließend auf einen Webserver im Internet hochlade. Wie erwartet, bleibt der Vorgang unbemerkt. Aus Zeitgründen muss ich auf eine weitere Eskalation der Berechtigung verzichten – möglicherweise wird es dafür einen weiteren Penetrationstest geben.

Nachklang

Ich rufe meinen Auftraggeber an und informiere ihn mündlich über die gravierendsten Sicherheitslücken. Er zeigt sich

ernüchtert vom Resultat meines Penetrationstests. Weil ich keine Angriffswerkzeuge oder Schadsoftware eingesetzt habe, waren die im Unternehmen eingesetzten Sicherheitslösungen gegen meinen Angriff schlussendlich wirkungslos. Am Ende war es – wie so oft – menschliches Versagen, das mir ein Einfallstor in die geschützte Infrastruktur meines Auftraggebers eröffnete. Mehrere Mitarbeiter fielen auf meine Phishing-Attacke herein und die fehlende Multi-Faktor-Authentifizierung bei einem von drei Fernzugriffsportalen führte dazu, dass ich die erbeuteten Zugangsdaten tatsächlich nutzen konnte, um ins System einzubrechen. Eine alte Weisheit lautet „Der Angreifer muss nur einmal gewinnen, der Verteidiger immer“ – und dieser Black-Box-Test hat einmal öfter gezeigt, dass etwas Wahres dran ist.

Abgeschlossen ist die Geschichte an dieser Stelle allerdings weder für mich noch für meinen Auftraggeber. Zu meinen Aufgaben als Pentester gehört es auch, am Ende des Pentests einen aussagekräftigen Abschlussbericht zu erstellen. Notgedrungen setze ich mich wieder an den Schreibtisch und fasse die durchgeführten Schritte und die Ergebnisse des Black-Box-Tests zusammen. Ich beschreibe die entdeckten Schwachstellen genau und versuche, mich dabei möglichst verständlich auszudrücken. Das Ziel ist es schließlich, dass mein Auftraggeber die von mir entdeckten Schwachstellen versteht und sie beseitigen kann. *(kst@ct.de)*

Alle erwähnten und verwendeten Werkzeuge: ct.de/yhxe