Deception-basierte Verteidigung mit Canarytokens

Eine noch junge Disziplin im Kampf gegen Cyberangriffe ist Deception (Täuschung). Mit dieser offensiven Methode legen Verteidiger falsche Spuren oder Fallen aus, mit denen sie Angreifer entdecken, deren Vorgehensweisen beobachten und im günstigsten Fall den Angriff ausbremsen oder abwehren können.

Von Gian-Luca Buol

Im Kontext von Cyberangriffen setzen Angreifer viele Techniken ein, die darauf abzielen, die Erkennungsmethoden von Sicherheitslösungen zu umgehen. Dabei reicht es aus, eine Technik zu nutzen, die das Produkt nicht als bösartig einstuft. Signaturbasierte sowie verhaltensbasierte Ansätze sind anfällig für Fehlalarme. Das führt dazu, dass Verteidiger zu einem gewissen Grad abstumpfen. Um diese Schwächen herkömmlicher Methoden auszugleichen, haben sich Deception-Technologien (DT) als ergänzender Ansatz etabliert [1].

Deception, ein aus dem Englischen stammender Begriff, der Täuschung bedeutet, basiert auf dem Konzept, Angreifer mithilfe sogenannter Lures – also Lockmittel oder Fallen – anzuziehen und zu einer Interaktion zu verleiten. Dieser Ansatz ermöglicht das Erkennen von Eindringlingen mit einer geringen falsch positiven Rate und erfordert lediglich einen minimalen Wartungsaufwand.

iX-TRACT

- Cyber Deception ersetzt zwar keine anderen Sicherheitsmechanismen, ist aber für Verteidiger eine wertvolle Ergänzung dazu.
- Mit Canarytokens lassen sich effektiv und preiswert hochwertige Signalquellen für das Erkennen von Cyberangriffen einrichten.
- Der strategische Einsatz von Deception ermöglicht es Unternehmen, Angreifer frühzeitig zu identifizieren und wertvolle Informationen über ihre Taktiken und Verfahren zu gewinnen.

Cyberangriffe entdecken

Im Jahr 2021 erkannten die meisten Unternehmen einen Cyberangriff noch selbst durch interne Detektionsmechanismen. Im Folgejahr wurden rund 63 Prozent der Unternehmen von Externen darauf aufmerksam gemacht, dass sie derzeit angegriffen werden oder bereits erfolgreich kompromittiert wurden. Dies zeigt der M-Trends-2023-Bericht von Mandiant (alle im Text genannten Quellen siehe ix.de/zks2). Ein Lichtblick zeichnet sich dafür bei der Verweildauer ab, also wie lange sich Angreifer unbemerkt in der Infrastruktur ihrer Opfer bewegen. Hier sank der Median von 21 auf 16 Tage. Diese Verbesserung belegt eine schnellere Reaktion auf Cyberangriffe als zuvor.

Dennoch haben es Eindringlinge aufgrund der Asymmetrie zwischen Angriff und Verteidigung immer noch einfacher: Angreifer brauchen nur eine einzige ausnutzbare Schwachstelle zu finden, die Verteidiger dagegen müssen alle Schwachstellen entdecken und stopfen. John Lambert, ehemaliger Manager des Microsoft Threat Intelligence Center, bemerkte in seinem Blogartikel „A Defender's Mindset" (siehe ix.de/zks2) zutreffend, dass sich das auf unterschiedliche Denkweisen zurückführen lässt [2].

So arbeitet die IT-Industrie häufig nach Checklisten. Das zeigt sich auch in der Verteidigung: Sie pflegt Listen mit Schwachstellen oder Datenbanken mit Signaturen schädlicher Software. Angreifer denken hingegen in Graphen und sehen ihr Ziel als eine Menge von Knoten und Kanten. Diese repräsentieren die verschiedenen IT-Systeme und ihre Schnittstellen. Deshalb ist es für Verteidiger zentral, regelmäßig eine Angreiferperspektive einzunehmen, um mögliche Angriffspfade zu entdecken. Dies trifft auch auf Deception zu. Lures sollte man so entwerfen, dass sie für Angreifer interessant genug wirken, um mit ihnen zu interagieren, sie aber zugleich nicht sofort enttarnen – eine Wanderung auf einem schmalen Grat.

Der Deception-Dschungel

In der Praxis verwenden verschiedene Publikationen und kommerzielle Anbieter von Deception-Lösungen Bezeichnungen inkonsistent; oft vermischen sie existierende Begriffe oder erfinden ganz neue Namen. Für ein besseres Verständnis werden im Nachfolgenden die zentralen Bezeichnungen erläutert.

Cyber Deception und Honeypots, im Deutschen auch als Honigtöpfe bezeich-

Schematische Darstellung einer bösartigen Interaktion mit einer tokenisierten Ressource: Der Angreifer weiß nicht, dass seine Aktion einen Alarm auslöst (Abb. 1).

net, werden synonym als Oberbegriff für Deception-Technologien verwendet. Der Begriff „Honigtöpfe“ ist missverständlich, da sie oft mit einem Computer oder einer physischen Ressource assoziiert werden, mit der ein Angreifer interagieren kann. Das war zwar die ursprüngliche Erscheinungsform von Honeypots, trifft heute jedoch nicht mehr zu. Stattdessen kann es sich bei einem Honigtopf um eine beliebige digitale Ressource handeln. Das kann eine Kreditkartennummer, eine Excel-Tabelle, ein Datenbankeintrag oder ein Benutzeraccount sein. Honeypots gibt es in vielen Formen und Größen, aber sie alle haben das gleiche Konzept: eine digitale Ressource, deren Wert in ihrer unbefugten Nutzung liegt.

Honigtöpfe lassen sich in drei wesentliche Kategorien unterteilen. Ein Honey-System stellt ein voll funktionsfähiges System inklusive Betriebssystem und seiner Komponenten dar. Jegliche Kommunikation mit dem System wie auch die Aktivitäten darauf werden überwacht. Der Honey-Service ahmt die Funktion einer Software oder eines Protokolls nach, beispielsweise einen SSH-Dienst. Eine Sammlung an Honey-Services kann auch ein ganzes System imitieren. Honey-Token bilden die kleinste Einheit und ahmen legitime Daten nach, etwa Dokumente oder Datenbankeinträge. Sobald sie geöffnet oder abgefragt werden, löst das einen Alarm aus. Diese drei Formen werden in der Praxis oft noch weiter unterteilt, etwa in Honeyfiles, also präparierte Dateien. Für eine konsistente Verwendung empfiehlt sich das Glossar der Non-Profit-Organisation MITRE aus ihrem Engage-Framework (siehe ix.de/zks2).

Angreifer entdecken, ausforschen und ablenken

Die Literatur, etwa ein Forschungsdokument der französischen Telekom, spricht von Threat Research, Intrusion Detection und Resource Exhaustion (siehe ix.de/zks2). Diese drei sind die dominierenden Einsatzzwecke von Deception. Die Angreiferanalyse (Threat Research) nutzt Honeypots, um Techniken und Werkzeuge unterschiedlicher Bedrohungsakteure zu erforschen und zu dokumentieren. Die Ergebnisse fließen dann in Datenbanken zu Cyber Threat Intelligence (CTI) ein.

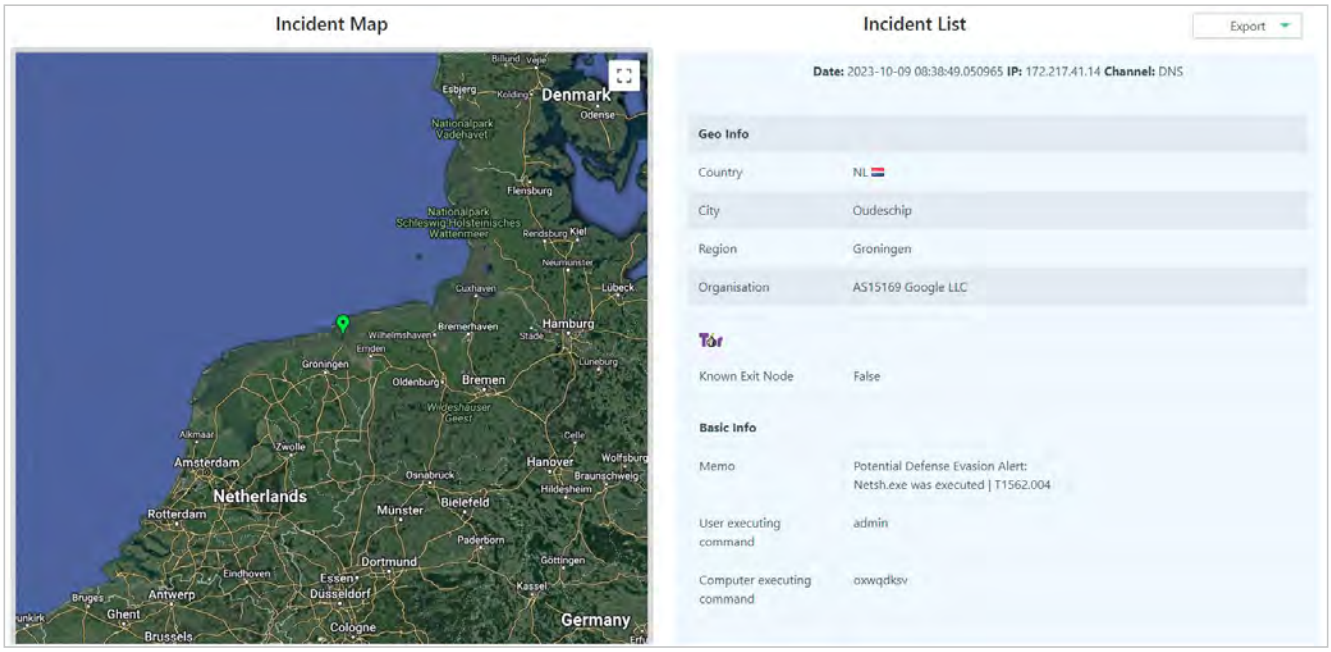
Wenn ein Angreifer ein Netzwerk kompromittiert, muss er irgendwann mit den enumerierten Ressourcen im Netzwerk interagieren [3]. Intrusion-Detection-Honeypots zielen auf diese Interaktionen ab. Eine einzelne Interaktion genügt in der Theorie bereits, damit der Eindringling entdeckt wird. Beim dritten Einsatzzweck, der Ablenkung, sollen zahlreiche Honigtöpfe mit ausreichender Interaktion möglichst viele Ressourcen und damit Zeit des Angreifers binden. Das verschafft der Verteidigung mehr Zeit.

Diese drei Zwecke schließen sich nicht gegenseitig aus. Beim Implementieren von Honeypots kann man einzelne oder auch alle drei Ziele zur selben Zeit verfolgen. Abhängig vom Einsatzszenario konfiguriert man die Honigtöpfe unterschiedlich. Ausschlaggebend sind dabei ihre vier Hauptcharakteristiken.

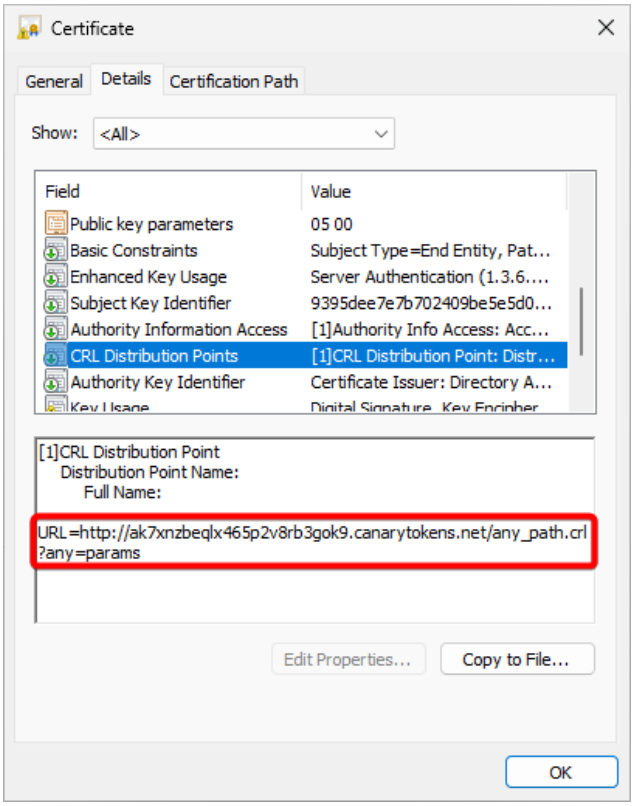
Was macht Honeypots aus?

Honeypots sind täuschend, auffindbar, interaktiv und werden überwacht. Deception besteht meist aus zwei Teilen: dem Verbergen des Realen und dem Zeigen des Fiktiven. Angewandt auf Honeypots gilt es also, diejenigen Merkmale zu verbergen, die einen Honeypot als solchen identifizierbar machen, und die Merkmale der Ressource hervorzuheben, die der Honigtopf vortäuscht zu sein.

Ein Honeypot hat nur dann einen Wert, wenn er auch gefunden wird und



Ein Webportal nennt weiterführende Details zu einem ausgelösten Canarytoken, etwa Informationen zur IP-Adresse (Abb. 2).



Das Canarytoken ist als CRL in einem X.509-Zertifikat hinterlegt und schlägt beim Aufrufen der Rückrufliste Alarm (Abb. 3).

eine Interaktion stattfindet. Honigtöpfe werden immer in einem Kontext platziert. Außerhalb dieses Kontextes sollten keine Interaktionen möglich sein, um die Anzahl der Fehlalarme gering zu halten.

Der Erfolg eines Cyberangriffs hängt maßgeblich von den Fähigkeiten des Angreifers ab, sein Ziel zu erkunden, Eintrittswege zu finden und seine Rechte auf den Zielsystemen zu erweitern. Dann wird er unweigerlich mit einer Ressource interagieren müssen, um mehr Informationen über sie zu sammeln oder sie anzugreifen. Das Interaktionslevel des Honigtopfs spielt eine wichtige Rolle. In der Praxis unterscheidet man zwischen Low-Interaction- und High-Interaction-Honeypots. Das sind jedoch keine statischen Kategorien, sondern zwei Enden eines Spektrums, in dem sich die verschiedenen Formen von Honeypots einordnen lassen. Je mehr Interaktion ein Honigtopf zulässt, desto realistischer fühlt sich die Ressource für den Angreifer an und er verbringt potenziell mehr Zeit damit.

Findet eine Interaktion mit einem Honigtopf statt, muss sie aufgezeichnet werden; ansonsten ist der Honeypot wertlos. Abhängig vom Interaktionslevel und dem Einsatzzweck unterscheiden sich die Methoden für die Überwachung.

Letztlich muss man alle vier Charakteristiken sorgfältig auswählen und auf das Einsatzszenario abstimmen. Andernfalls besteht die Gefahr, dass keine Interaktionen stattfinden oder umgekehrt – es finden zu viele ungewollte Interaktionen statt, was die Aussagekraft der Alarme stark mindert. Doch wie können Sicherheitsverantwortliche und Admins Cyber Deception praktisch einsetzen?

Canarytokens von Thinkst

Canarytokens ist eine Deception-Lösung, die als kostenfreier Webdienst vom Hersteller Thinkst angeboten wird, aber auch unter einer freien Lizenz auf GitHub verfügbar ist (ix.de/zks2). Sie ist nicht zu verwechseln mit Stack Canaries, einer Art Prüfsumme, die als Schutzmechanismus gegen Buffer-Overflow-Angriffe dient. Obwohl mittlerweile eine beachtliche Zahl an Securityherstellern auf den Deception-Zug aufgesprungen sind, stechen die Canarytokens aus zwei Gründen hervor: Die Lösung ist Open Source und ihr modularer Aufbau ermöglicht einen flexiblen Einsatz. Canarytokens können verschiedene Formen annehmen. Thinkst bietet diverse vorpräparierte Token an, beispielsweise eingebettet in Word-Dokumente, Windows-Registry-Schlüssel oder auch in benutzerdefinierte EXE- oder DLL-Dateien (ix.de/zks2).

Der Funktion sämtlicher Canarytokens liegen zwei elementare Internetprotokolle zugrunde: HTTP-GET- und DNS-Anfragen. Sowohl HTTP- wie auch DNS-Token lassen sich beliebig in eine Datei, eine Webseite, eine Datenbank oder in ein System einbauen. Dieses modulare Design macht die Canarytokens so effektiv und vielseitig einsetzbar, wie ein Artikel auf dem Cybersecurity-Blog amartinsec zeigt (siehe ix.de/zks2). Jedes generierte Token ist eindeutig und löst beim Aufruf einen Alarm aus (siehe Abbildung 1).

Für das Erzeugen von Canarytokens können Verteidiger die Onlineinfrastruktur von Thinkst verwenden oder die Infrastruktur selbst betreiben. Dazu stellt Thinkst das Projekt als Docker-Image zum Download bereit (ix.de/zks2). Der Eigenbetrieb ermöglicht die Konfiguration weiterführender Parameter: Beispielsweise kann man die eigene Domäne verwenden, was es für Angreifer schwieriger macht, zwischen einer produktiven und einer Deception-Ressource zu unterscheiden.

Verteidiger können auch die Alarmierung auf die eigene Organisation abstimmen, indem sie die ausgelösten Alarme zu Token direkt an ihr SIEM-System (Security Information and Event Management) weiterleiten. Setzen sie hingegen die Infrastruktur von Thinkst ein, verwenden alle Token die Domäne canarytokens.com und die Alarmierung erfolgt über E-Mail. Die Mail enthält einen Link, der zu einem Webportal mit weiterführenden Informationen führt (siehe Abbildung 2).

Das Beispiel zeigt ein Ereignis zur Ausführung von `netsh.exe`, einem vorinstallierten Windows-Dienstprogramm, das Angreifer oft missbrauchen. Thinkst erfasst verschiedene Details zur öffentlichen IP-Adresse sowie Benutzername und Hostname, unter dem der Befehl ausgeführt wurde. Dies ist besonders dann interessant, wenn in einer Organisation Daten exfiltriert und später aus einem anderen Land geöffnet werden, in dem das bestohlene Unternehmen nicht tätig ist.

Oft lässt sich die IP-Adresse nicht direkt mit den Eindringlingen in Verbindung bringen, wenn diese ein Anonymisierungsnetzwerk nutzen. In diesem Beispiel wird das Token über eine DNS-Abfrage ausgelöst. HTTP-Token sind vielseitiger, da das Protokoll mehr Optionen im Header bietet. Dann kann Canarytokens noch mehr Datenpunkte aufzeichnen, etwa User Agent, Browser-Ver-

Listing 1: Eingebettetes Canarytoken in der Verzeichnisstruktur eines Word-Dokuments

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships
        xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
        <Relationship Id="rId1" Type="http://schemas.openxmlformats.org/↵
                        officeDocument/2006/relationships/image" Target=↵
            "http://canarytokens.com/traffic/about/ujzjn2vrglorf1z8om87xq7hj/↵
                                        index.html" TargetMode="External"/>
</Relationships>
```

Das kaum erkennbare Canarytoken im Word-Dokument fungiert als Tracking-Pixel (Abb. 4).

sion oder Sprache des Browsers. In der Praxis geht der HTTP-Verkehr jedoch meist gefiltert ins Internet – DNS hingegen meist nicht. Thinkst ermöglicht es, beim Erstellen je nach Token beide Protokolle gleichzeitig einzusetzen.

Mit Zertifikat: Custom Binary Token

Custom EXE Token, auch bezeichnet als Binary Token, sind eine besonders interessante Kreation von Thinkst. Dieses Token funktioniert, indem eine ausführbare Datei (EXE) oder eine Dynamic Link Library (DLL) mit einem Zertifikat signiert wird, das das Canarytoken beinhaltet. Wenn die EXE ausgeführt oder die DLL geladen wird, löst das einen Alarm aus.

Das Zertifikat enthält das Canarytoken an zwei Stellen: Erstens wird das Token als Subject Alternative Name (SAN) eingefügt. Der SAN ist ein optionaler Parameter in einem X.509-Zertifikat, der es erlaubt, dem Zertifikat neben der Hauptdomäne zusätzliche Domänennamen hinzuzufügen. Zweitens wird das Token auch als Certificate Revocation List (CRL) Distribution Point hinterlegt (siehe Abbildung 3). Eine CRL ist eine Liste von Zertifikaten, die vor ihrem Ablauf widerrufen wurden. Dies geschieht beispielsweise, wenn eine Zertifizierungsstelle kompromittiert wurde.

Wenn eine signierte Software startet, verifiziert das Betriebssystem die Echtheit des Zertifikats. Das Auflösen des SAN oder auch die Abfrage der CRL führt dazu, dass das Token und damit der Alarm ausgelöst wird.

Microsoft Word Token

Canarytokens in der Form von Word-Dokumenten alarmieren, sobald sie mit Microsoft Word geöffnet werden. Wird das Dokument mit einem Texteditor oder anderen Programmen angezeigt, löst dies keinen Alarm aus. Analog funktioniert das Token für Excel-Tabellen.

Microsoft-Office-Dokumente, die eine Dateiendung mit *x haben, sind eigentlich komprimierte Archive. Innerhalb jeder Office-Datei gibt es eine Verzeichnisstruktur mit mehreren XML-Dateien, die die verschiedenen Teile des Dokumentes repräsentieren: Beispielsweise gibt es XML-Dateien für Kopf- und Fußzeilen. Die Summe aller XML-Dateien bildet die Struktur des Dokuments. Die Beziehung zwischen den XML-Dateien und einer Office-Datei wird durch die Open Packaging Conventions (ix.de/zks2) definiert. Thinkst platziert das Canarytoken in der Fußzeile des Word-Dokumentes. Es kann mit einem Archivtool wie 7-Zip extrahiert werden. Das Token findet sich nach dem Entpacken in einer XML-Datei unter dem relativen Pfad ./word/_rels/footer2.xml.rels. Der Inhalt der Datei sieht aus wie in Listing 1.

Das Token wird als Bild in der Fußzeile mit der Feldfunktion INCLUDEPICTURE eingebettet. Feldfunktionen haben die Aufgabe, Inhalte in Word dynamisch zu aktualisieren, zum Beispiel Datum oder Seitenzahl. INCLUDEPICTURE bewirkt das Nachladen eines Bildes von der im XML spezifizierten Quelle. In diesem Fall ist dies das Token selbst, das auslöst, sobald das Dokument geöffnet wird und Word das Bild zu laden versucht. Im geöffneten Dokument ist das geladene Bild nur bei starker Vergrößerung und Markieren der entsprechenden Stelle in der Fußzeile schwach sichtbar (siehe Abbildung 4).

Sensitive Command Token

Mit dem Sensitive Command Token lässt sich der Start von Programmen beziehungsweise Prozessen erkennen. Dabei ist es gleich, ob das im Token hinterlegte Programm auf dem Zielsystem zum Zeitpunkt des Implementierens schon vorhanden ist oder ein Angreifer es erst später auf das Zielsystem hochlädt. In der Praxis eignen sich hierfür besonders Windows-Bordmittel, die von Angreifern oft missbraucht werden; diese Technik nennt sich Living Off The Land (LOL oder LOTL). Eine gute Quelle für solche internen Dienstprogramme ist das ATT&CK-Framework von MITRE [4]. Eine aktuelle Aufstellung typischer LOLBins und wie sie genutzt werden, bietet das LOLBAS-Project (siehe ix.de/zks2).

Das Sensitive Command Token implementiert man durch zwei Einträge in der Registrierungsdatenbank (siehe Listing 2). Seine Funktion wird im Folgenden anhand des Programms `netsh.exe` erklärt. Der erste Registrierungsschlüssel wird unter den Image File Execution Options (IFEO) angelegt. Windows verwendet IFEO für das Debuggen von Prozessen mithilfe von Tracing Flags, die an dieser Stelle in der Registrierung stehen. Im Kontext des Canarytoken wird ein globales Kennzeichen gesetzt, das eine Ausnahme wirft, sobald das Programm `netsh.exe` ausgeführt wird. Durch diesen Eintrag wird der Start des Programms erkannt.

Der zweite Registrierungsschlüssel richtet eine Art Überwachungsprozess ein, der ausgelöst wird, wenn das Programm wieder beendet wird. Sobald dieser Fall eintritt, führt Windows im Hintergrund eine DNS-Anfrage auf das zuvor erzeugte Canarytoken aus.

Listing 2: Registrierungskonfiguration eines Sensitive Command Token für netsh.exe

```
; das zu überwachende Programm
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File ↵
                                                    Execution Options\Netsh.exe]
"GlobalFlag"=dword:00000200

; eindeutiges Canarytoken, das den Alarm auslöst, wenn der Prozess beendet wurde
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows ↵
                                    NT\CurrentVersion\SilentProcessExit\Netsh.exe]
"ReportingMode"=dword:00000001
"MonitorProcess"="cmd.exe /c start /min powershell.exe -windowstyle hidden –↵
          command \"$($u=$(\\\"u$env:username\\\" -replace('[^a-zA-Z0-9\\-]+',↵
  ''))[0..63] -join '';$c=$(\\\"c$env:computername\\\" -replace('[^a-zA-Z0-9\\-↵
                                    ]+', ''))[0..63] -join '';Resolve-DnsName -Name↵
            \\\"$c.UN.$u.CMD.1hk3km5d51sg0xvx3t2fb4pba.canarytokens.com\\\")\""
```

Ein PowerShell-Prozess löst das Token aus und ermittelt über die Umgebungsvariablen zusätzlich den aktuellen Benutzer und den Hostnamen und sendet diese Informationen mit. Sie sind, wie in Abbildung 2 ersichtlich, im Portal von Canarytokens dargestellt.

Eine Einschränkung des Sensitive Command Token liegt darin, dass das hinterlegte Token ausschließlich auf den Namen des zu überwachenden Prozesses reagiert. Wurde beispielsweise ein Token für `mimikatz.exe` angelegt [5], ein Angreifer hat das Programm zum Auslesen von Anmeldeinformationen aber in `explorer.exe` unbenannt, wird das Token nicht auslösen. Eine weitere Restriktion liegt darin, dass der Prozess ordnungsgemäß beendet werden muss. Falls er beispielsweise durch einen Signal Interrupt (Ctrl + C) abgebrochen wird, wird die Funktion `Silent ProcessExit` aus dem Registrierungsschlüssel nicht angewendet und infolgedessen das Token nicht ausgelöst. Details dazu beschreibt ein Blogartikel von Casey Smith (siehe ix.de/zks2).

Die Onlinedokumentation von Canarytokens (ix.de/zks2) beschreibt weitere spannende Token, die beispielsweise bei SQL-Datenbankabfragen oder beim Aufrufen einer Website auslösen.

Größter Nutzen von Deception

Deception-basierte Detektionsansätze haben einen entscheidenden Vorteil gegenüber traditionellen Sicherheitsansätzen, wie der „Implementer's Guide for Deception Technologies" des SANS Institute darlegt (ix.de/zks2). Sie berücksichtigen technologische Trends sowie die sich weiterentwickelnden Taktiken, Techniken und Verfahren (TTP) der Bedrohungsakteure. Obwohl keine Technik eine hundertprozentige Erkennungsgarantie bietet, erleichtert Deception das Erkennen vieler Angriffspfade erheblich.

Das beruht auf einem einfachen, aber effektiven Prinzip: Ein Angreifer, der erfolgreich in eine Umgebung eindringt oder eine Identität kompromittiert, hat in der Regel nur begrenzte oder keine Kenntnisse über die spezifischen Merkmale seines Ziels und muss es zunächst erkunden. Bei einer guten Täuschungsstrategie gleichen die falschen Ressourcen den echten bis ins Detail. Gibt es ausreichend Täuschungsressourcen in der Umgebung, wird es wahrscheinlich, dass ein Angreifer mit mindestens einer von ihnen interagiert.

Die Anzahl allein ist jedoch nicht das einzige relevante Kriterium, wenn es um die Wirksamkeit der Detektion geht. Auch die Genauigkeit ist entscheidend. Traditionelle Erkennungstechnologien streben in der Regel ein ausgewogenes Verhältnis zwischen der Rate von Falsch-positiv- und Falsch-negativ-Ereignissen an. Zu viele falsch negative Ergebnisse sind ein Problem, da dadurch Angreifer länger unentdeckt bleiben. Gleichzeitig erzeugen viele falsch positive Ergebnisse ein Grundrauschen, das die Verteidigung ablenkt.

Ein weiterer Unterschied: Produkte für Endpoint Detection and Response (EDR) [6] wie auch ein Antivirus (AV) sind zu einem gewissen Grad voreingenommen, was eine legitime Aktion ist und was nicht. Signaturbasierte Ansätze funktionieren binär: Eine Aktion ist bösartig, weil sie in der Vergangenheit bereits einmal als bösartig eingestuft wurde; alles andere wird automatisch für gutartig befunden. Verhaltensbasierte Ansätze beruhen auf Modellen, die mit maschinellem Lernen auch unbekannte Bedrohungen identifizieren können. Beide Verfahren bewerten jedoch, ob eine Datei oder ein Prozess bösartig ist oder nicht. Das Ergebnis bestimmt dann die Aktion, die ausgelöst wird. Diese Voreingenommenheit schränkt die Verfahren ein. Bei Deception tritt sie nicht auf, denn hinter einem Honeypot steckt keine Logik, die evaluiert, ob eine Aktion bösartig ist. Jede Interaktion löst einen Alarm aus.

Deception bringt den größten Nutzen, wenn Verteidiger die Honigtöpfe bereits vor einem Angriff ausgelegt haben. Meine Praxistests im Rahmen einer wissenschaftlichen Arbeit (ix.de/zks2) haben gezeigt, dass Eindringlinge insbesondere in der ersten Angriffsphase, der Informationssammlung, am meisten Alarme auslösen. Jedoch auch während eines Angriffs kann es noch nützlich sein, Deception einzusetzen. Denn Fallen, in die der Angreifer getappt ist, können der Verteidigung wertvolle Informationen über ihn und sein Vorgehen liefern. Aktuelle Analysen von Cyberangriffen zeigen zudem, dass Angreifer regelmäßig EDR- oder AV-Produkte deaktivieren. Hier können Lures als eine ergänzende Detektionsschicht dienen, wenn die primäre Sicherheitslösung keine Alarme mehr liefert. Im Ernstfall sollte die Reaktion jedoch mit einem Incident-Response-Experten abgestimmt werden.

Fazit

Deception-Technologien läuten einen Paradigmenwechsel in der Denkweise aufseiten der Verteidiger ein. Das kann helfen, Lücken in traditionellen Erkennungsmechanismen zu schließen und den aktuellen Trends, wie von Mandiant beobachtet, entgegenzuwirken. Die Beispiele in diesem Artikel zu einigen ausgewählten Canarytokens zeigen, dass mit geringem Aufwand und vergleichsweise wenig Geld hochwertige Signalquellen entstehen können.

Deception ist jedoch kein vollständiger Ersatz für bestehende Sicherheitsansätze, sondern ergänzt und verbessert sie. Neben guten Detektionsmechanismen gehören weitere Punkte zu einer ganzheitlichen Sicherheitsstrategie für Organisationen: Hier sind vor allem Patch- und Schwachstellenmanagement, Multi-Faktor-Authentifizierung, Active-Directory-Härtung oder das Least-Privilege-Prinzip zu nennen. Nur auf einem soliden Fundament lassen sich Deception-Technologien sinnvoll betreiben. (ur@ix.de)

Quellen

[1] Frank Ully; Überlistet und ausgebremst; Active Directory – wie Angreifer mit Deception in die Falle gelockt werden; iX 11/2021, S. 120

[2] Frank Ully; Gefühlt sicher: Sicherheitsmythen und -irrtümer; iX 6/2023, S. 44

[3] Frank Ully; Nach oben gehangelt; Informationsbeschaffung – was jeder Domänenbenutzer alles sieht; iX 10/2020, S. 58

[4] Martin Karl Junghans, Joshua Ziemann; Gewusst wo; Raus aus dem Maßnahmensumpf: eine Anleitung zum Emotet-Selbsttest; iX 2/2021, S. 42

[5] Frank Ully; Fette Beute; Passwörter und Hashes – wie Angreifer die Domäne kompromittieren; iX 11/2020, S. 94

[6] Konstantin Bücheler, Martin Hartmann, Alain Rödel, Stefan Strobel; Auf dem Radar; Endpoint Detection and Response: Gefahren schnell erkennen und reagieren; iX 11/2021, S. 52

[7] Alle im Text erwähnten Werkzeuge, Dokumentationen und Blogartikel sind über ix.de/zks2 zu finden.

GIAN-LUCA BUOL

ist Information Security Officer bei der V-ZUG AG. In seiner Bachelorthesis entwickelte er für die Oneconsult AG einen Deception-Prototyp für den Einsatz in Incident-Response-Fällen.