Red Teaming: Post Exploitation und Lateral Movement

Seitwärts-bewegungen

Sascha Herzog

Das Eindringen in ein IT-Netzwerk ist die eine Sache, sich darin zu bewegen und Kontrolle über wichtige Teile davon zu erlangen, eine andere. Erkenntnisse darüber, wie ein Angreifer vorgeht, sind für die Verteidigung zwingend erforderlich.

iX-TRACT

- Wenn Angreifer Systeme kompromittieren, müssen sie, um Daten stehlen oder manipulieren zu können, nicht gleich an der richtigen Stelle landen oder entsprechende Rechte besitzen. Es genügt erst einmal, „drinnen“ zu sein.
- Lateral Movement nennt sich die Strategie von Angreifern, sich nach dem Eindringen in ein Netzwerk in verschiedenen Bereichen zu bewegen und nach und nach die für die geplanten Aktionen erforderlichen Rechte zu erlangen.
- Sensibilisierung von Mitarbeitern sowie eine sinnvolle Netzwerksegmentierung und Kontrolle der Zugänge und Verbindungen zählen zu den wichtigsten Strategien, mit denen man das Sichausbreiten eines Angreifers im Netzwerk unterbindet oder zumindest erschwert.

Der sechste Artikel der Serie zu Red Team Assessments beschäftigt sich mit einer der komplexesten und wichtigsten Phasen einer solchen Übung, die auch bei wirklichen APT-Angriffen (alle Abkürzungen siehe Glossar) für die Angreifer- sowie die Verteidigerseite kriegsentscheidend ist: Post Exploitation und Lateral Movement. Post Exploitation bezeichnet die Aktionen, die unmittelbar nach dem Eindringen in ein System stattfinden, unter Lateral Movement versteht man das Sichbewegen nach allen Seiten in einem Netzwerk.

In dieser Phase hat man es als Angreifer bereits geschafft, den Perimeter, also die äußeren Schutzmauern seines Ziels, zu überwinden, und besitzt Zugang zu mindestens einem internen Netzbereich (beschrieben in den vorherigen Artikeln, siehe *iX* 2/2018, 4/2018, 6/2018, 9/2018, 10/2018). Die Aufgabe besteht nun darin, sich innerhalb dieses Netzwerks fortzubewegen, um weitere, strategisch relevante Netzkomponenten wie Server, Admin-Workstations, Datenbanken, Netzwerkkomponenten et cetera unter seine Kontrolle zu bekommen.

Dies ist meistens notwendig, um letztendlich eine „kritische Funktion“ zu kompromittieren, die wiederum zentrale Assets des Ziels stark beeinträchtigen oder gänzlich zerstören könnte. Dieser Begriff – „Critical Function“ – stammt aus dem im Mai 2018 von der europäischen Zentralbank EZB veröffentlichten Rahmenwerk für ethisches Hacking auf der Basis von Threat Intelligence (das Framework und alle weiteren Links des Artikels sind über ix.de/ix1812082 zu finden). Es beschreibt kontrollierte und individuell zugeschnittene Tests in Bezug auf Cyberangriffe auf den Finanzmarkt.

Ein Lateral Movement unterscheidet sich stark von einem internen Penetrationstest. Bei Letzterem werden meist Systeme, Ports und Services gescannt und auf Schwachstellen hin geprüft, die die Pentester dann unter Umständen ausnutzen können (Exploitation). Ein interner Pentest ist infolgedessen sehr „laut“ und mit der internen IT abgestimmt, teilweise findet er sogar in Testumgebungen statt.

Heimlich, still und leise

Bei einem Lateral Movement indessen, das immer in produktiven Umgebungen durchgeführt wird, muss man darauf achten, keine Alarmglocken auszulösen und unter dem Radar zu bleiben („OPSEC-Safety“). Statt zu scannen und Dienste zu exploiten, verwendet man die Bordmittel

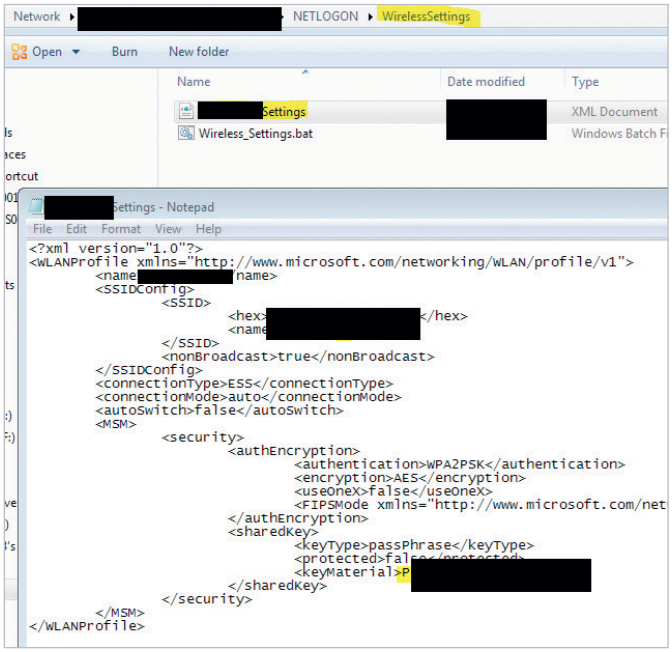
```
      </TapiUnattendLocation>
    </component>
  - <component name="Microsoft-Windows-SystemRestore-Main" processorArchitecture="x86" publicKeyToken="31bf3856ad364e35" language="neutral"
      versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <DisableSR>1</DisableSR>
    </component>
  </settings>
- <settings pass="oo[redacted]
  - <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="x86" publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
      xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State">
    - <UserAccounts>
      - <AdministratorPassword>
          <Value>1[redacted]
          <PlainText>true</PlainText>
        </AdministratorPassword>
      - <LocalAccounts>
        - <LocalAccount wcm:action="add">
            <Description>Temp account</Description>
            <DisplayName>Temp account</DisplayName>
            <Group>Users</Group>
            <Name>TempAccount</Name>
          </LocalAccount>
        </LocalAccounts>
      </UserAccounts>
    - <AutoLogon>
        <Enabled>true</Enabled>
        <Username>Administrator</Username>
        <Domain>.</Domain>
      - <Password>
```

Die gefundene Datei half dabei, auf einigen lokalen Systemen Administratorrechte zu erlangen (Abb. 1).

des gekaperten Systems (oft eine MS-Windows-Workstation in einem Active Directory), arbeitet wenn möglich vom Arbeitsspeicher aus und vermeidet das Anlegen von Dateien sowie das Nachladen von Angreifer-Tools. Probate Mittel sind in AD-Umgebungen PowerShell, WMI, SMB- und DCOM-Kommunikation sowie das .NET-Universum. All das ist fast immer schon auf den Windows-Systemen vorhanden und erlaubt es Angreifern, wenige Spuren zu hinterlassen.

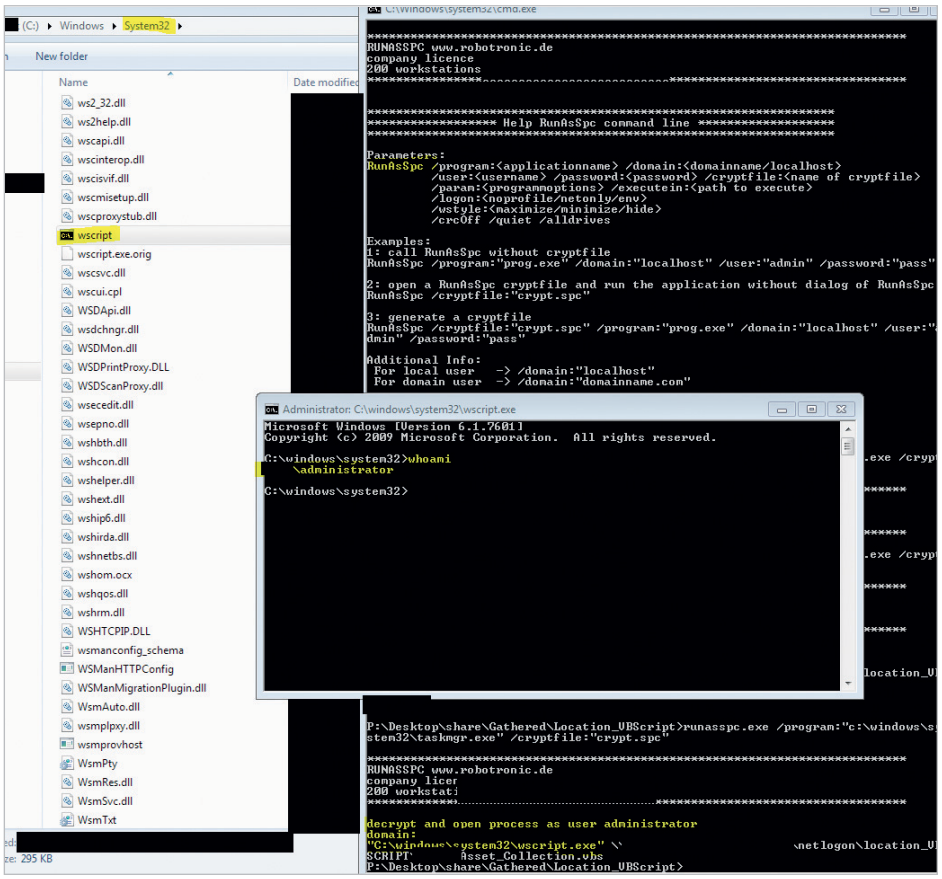
Im Folgenden soll ein Lateral Movement innerhalb eines Red Team Assessment beschrieben werden, das wir für einen Kunden durchgeführt haben. Zum Schutz des Kunden sind alle Daten anonymisiert und der Kontext leicht verändert.

Der Kunde stammt aus dem Bereich der Trinkwasserversorgung und der Entwässerung (Klärwerke) in der DACH-Re-



Plan B: Dank der ausgelesenen WLAN-Passwörter wäre bei Verlust der anderen Zugänge ein erneutes Eindringen in das Zielsystem möglich gewesen (Abb. 2).

gion. Er wollte in Erfahrung bringen, ob und wie es möglich ist, den kritischen Prozess der Wasserfiltrierung so zu manipulieren, dass entweder ungeklärtes Wasser zurück in Flüsse und Seen geleitet wird oder verschmutztes Wasser über die Trinkwasserversorgung unerkannt beim Verbraucher landet. Da beide Bereiche sehr komplex sind und aus diversen Schritten des Filterns, Messens und Weiterleitens bestehen, wird mittlerweile fast alles in diesem Zusammenhang über SCADA-Systeme zur Automatisierung der Abläufe und damit über IT-Systeme gesteuert.



Mit einigen Tricks ließ sich das Sicherheitskonzept aushebeln und eine Kommandozeile mit Administratorrechten für weitere Schritte erlangen (Abb. 3).

Unser Kunde nutzte die Wonderware System Platform, ein etabliertes System in diesem Bereich. Da es die mit Abstand kritischste Funktion innerhalb der Kundeninfrastruktur anbot, war es das erklärte Ziel, dieses System unter unsere Kontrolle zu bekommen, um damit den Filtrierungs- und Desinfektionsprozess stören zu können – ohne es tatsächlich zu tun.

Zugang über erbeutete Workstations

Nach einer längeren Phase der taktischen Informationsbeschaffung [1] und einem erfolgreichen Einbruch über einen damals neu erschienenen Firefox-Browser-Exploit auf drei Windows-Workstations, die sich im Office-Netzwerk des Wasserversorgers befanden, konnte die Post-Exploitation-Phase beginnen.

In einem solchen Fall – die Rede ist hier immer von kritischen produktiven Umgebungen – zieht es der Kunde in der Regel vor, dass wir diese Phase bei ihm vor Ort, meist ohne Wissen der IT-/OT-Abteilungen durchführen. Dabei steht uns immer mindestens ein eingeweihter leitender Mitarbeiter zur Verfügung, der die Umgebungen gut kennt. Durch dieses Vorgehen lassen sich Schadensfälle vermeiden, die durch das versehentliche Angreifen heikler Systeme oder Ressourcen entstehen können. Trotz des Einsatzes vor Ort arbeiteten wir wie echte Angreifer aus dem Internet über die drei gekaperten Systeme im internen Netz.

Oft gibt es in solchen Wasserwerken und ähnlichen ICS-Umgebungen, wie auch hier, keine eigene IT-Security-Abteilung, geschweige denn ein eigenes SOC-Team, was es Angreifern stark vereinfacht, sich lateral im internen Netz zu bewegen. Der IT-Leiter mit seinen zwei bis fünf Mitarbeitern ist für den Betrieb sowie für die Sicherheit der IT- und OT-Systeme verantwortlich.

Nachdem wir mit dem PowerShell Empire Framework unsere Kommando- und Kontrollverbindungen gesichert hatten und uns über zwei Wege eine Benutzerpersistenz auf den gekaperten Workstations verschaffen konnten, waren wir in der Lage, uns umzusehen und weiter vorzurücken. Persistenz in diesem Zusammenhang bedeutet, dass die Kommando- und Kontrollverbindung selbst nach einem Neustart oder Benutzer-Log-off/-Log-on wieder aufgebaut wird.

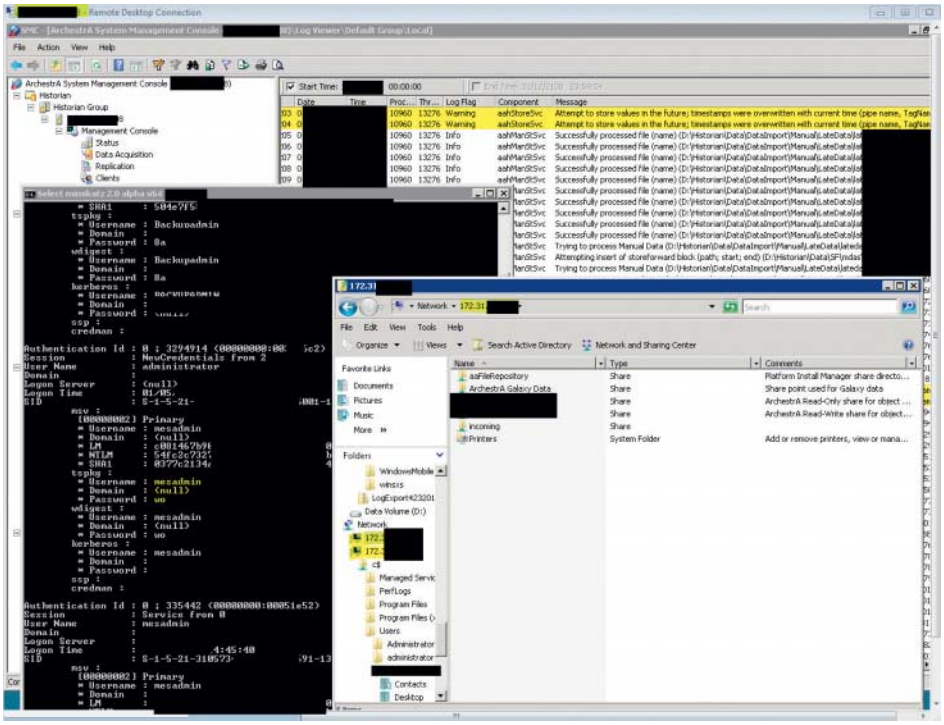
Wir begannen unsere Post Exploitation wie gewohnt mit einer Enumerierung der lokalen Systeme und Versuchen, adminis-

trative Rechte auf den Workstations zu erlangen. Dazu listeten wir alle relevanten Daten des Systems auf: beispielsweise Benutzer- und Gruppenzugehörigkeiten, sensible lokale Dateien und Registry Hives, laufende Dienste, unsichere Datei- und Verzeichnisrechte sowie Möglichkeiten, Prozesse über unsichere DLL-Ladevorgänge zu „entführen". Die grundlegenden Methoden dafür finden sich auf GitHub (siehe ix.de/ix1812082).

Bestand sichten, Wege ausloten

Nachdem keiner der Checks half, unsere Rechte auf einem lokalen System zu erhöhen, nutzten wir Methoden innerhalb des Netzwerks und der Windows-Domäne, in der sich die Workstations befanden. Wir suchten nach Passwörtern in Group-Policy-Preferences-Konfigurationen (Details siehe ix.de/ix1812082), enumerierten und durchsuchten alle erreichbaren Netzwerk-Shares, identifizierten ausführbare Dateien auf Shares, die mit unseren Benutzerrechten verändert werden konnten, und suchten weitere Möglichkeiten, entweder auf unseren Systemen oder auf einem entfernten System in der Domäne weitere Zugänge zu erlangen.

Wir wurden recht schnell auf einem Share fündig, das sehr wahrscheinlich für das Deployment von Systemen über das Windows Preinstallation Environment (Windows PE) verwendet wurde, und stießen auf eine *unattend.xml*-Datei mit dem in Abbildung 1 dargestellten Inhalt.



Mit den Rechten eines Domänenadministrators kann man sich via Remote Desktop an zahlreichen Systemen anmelden und sich dort weitere Rechte verschaffen (Abb. 4).

Listing: Port Forwarding

```
C:\>netsh interface portproxy add v4tov4 listenport=33899 listenaddress=172.39.120.98 ↵
                                                connectport=3389 connectaddress=172.39.119.65


C:\>netsh interface portproxy delete v4tov4 listenport=33899 listenaddress=172.39.120.98
```

Das brachte uns einen großen Schritt vorwärts, da wir uns somit auf einigen zugänglichen Windows Server- und Client-Systemen lokale Administratorrechte verschaffen konnten. Damit konnten wir über das Werkzeug Mimikatz (siehe ix.de/ix1812082) die Passwörter von eingeloggten Benutzern auf unseren Workstations auslesen – was leider nicht zu weiteren Benutzern führte.

Plan B: Übers WLAN

Außerdem konnten wir so auf entfernte System-Shares (C$), für die der lokale

Systeme mit mehreren Netzwerkkarten finden

Ein PowerShell-Einzeiler genügt, um die betreffenden Systeme aufzuspüren:

```
Get-WmiObject -Class Win32_Network ↵
   AdapterConfiguration -Filter IPEnabled= ↵
        TRUE -ComputerName <IP|Hostname of ↵
          target> | Format-Table -Property ↵
                      Description,IPAddress
```

Ein Resultat kann ungefähr so aussehen:

```
Description            IPAddress
----------             --------
Intel(R) PRO/1000      {172.16.68.18}
MT Network Connection

TAP-Windows Adapter V9 {192.168.7.22,fe80:
                       :5028:3fa5:4a07:6c0a}
```

Administrator galt, zugreifen und uns per *psexec.exe* auf diesen Systemen erhöhte Rechte verschaffen (nicht OPSEC-safe!) sowie lokale WiFi-Profile mit Klartextpasswörtern des WPA2-PSK-WLANs auslesen (Abbildung 2). Falls wir also in einer späteren Phase unsere C&C-Verbindungen verloren hätten, hätten wir als Notfallplan über das interne WLAN wieder einen Zugang erhalten können.

Innerhalb von drei Tagen sammelten wir auf diese Art weiter viele vertrauliche Daten und erlangten diverse Zugriffe auf interessante Systeme, zum Beispiel auf den Server zur Steuerung und Programmierung der Türschlossanlage auf dem gesamten Gelände.

Den Jackpot fanden wir allerdings am dritten Tag in einem Share, das uns nun über einen gekaperten Server zugänglich war. Hierin entdeckten wir ein Programm namens *runasspc* der Firma robotronic, das wir nicht kannten. Es stellte sich heraus, dass *runasspc* in Verbindung mit einer *crypt.spc*-Datei dazu verwendet wurde, hartcodierte Programme und Skripte mit den Rechten eines hardcodierten Benutzers und dessen hardcodiertem Passwort zu starten. In unserem Fall nutzte die IT-Abteilung es dazu, eine Asset-Sammlung auf allen Systemen der internen Windows-Domäne zu machen – was natürlich am besten mit Domain-Admin-Rechten, also den höchsten Rechten innerhalb eines Microsoft Active Directory, geht.

Da das aufgerufene Skript ein VB-Skript war, das über *wscript.exe* lief, konnten wir das Konzept leicht aushebeln und uns auf folgende Art Domain-Admin-Rechte verschaffen: Wir kopierten das *runasspc.exe*- und das *crypt.spc*-File auf eine Workstation, die unter unserer administrativen Kontrolle war, benannten die System-*cmd.exe* in *wscript.exe* um und starteten *runasspc*. Das Programm öffnete uns daraufhin eine Kommandozeile, die mit den Rechten des Domänenadministrators lief (Abbildung 3).

Die Suche nach Schnittstellen

Mit den neuen Rechten ausgestattet, waren wir in der Lage, alle Systeme im Netzwerk zu identifizieren, die mehr als eine Netzwerkkarte verwendeten (multihomed), um Netzwerkschnittstellen ins Prozessleitnetz zu finden. Der im Kasten „Systeme mit mehreren Netzwerkkarten finden“ gezeigte PowerShell-Einzeiler

Glossar

APT, Advanced Persistent Threat: komplexer, zielgerichteter Angriff.

C&C-Server, Command and Control: zentraler Server, der Befehle an gekaperte Rechner (Opfersysteme) verteilt und von diesen Informationen erhält.

DCOM, Distributed Component Object Model: von Microsoft definierte objektorientierte Interprozesskommunikation über ein Rechnernetz.

GPP, Group Policy Preferences: Gruppenrichtlinieneinstellungen, mit denen Administratoren auf Clients bestimmte Konfigurationen bereitstellen können.

ICS, Industrial Control Systems: industrielle Steuerungssysteme, die aus Soft- und Hardware sowie Vernetzungskomponenten bestehen.

ISMS, Information Security Management System: Managementsystem für Informationssicherheit, bestehend aus Prozessen und Regeln, die in einer Organisation die Informationssicherheit dauerhaft steuern, weiterentwickeln und verbessern sollen.

LAPS, Local Administrator Password Solution: Mit LAPS lässt sich pro Client ein dynamisches Passwort generieren und im Active Directory hinterlegen. Es soll das (unsichere) Hinterlegen lokaler Administratorpasswörter in den Gruppenrichtlinien ablösen.

MES, Manufacturing Execution System: eine prozessnah operierende Ebene eines mehrschichtigen Fertigungsmanagementsystems.

OT, Operational Technology: OT ist Hardware und Software, die eine Änderung durch die direkte Überwachung und/oder Kontrolle von physikalischen Geräten, Prozessen und Ereignissen im Unternehmen erkennen oder verursachen kann. Dazu gehören etwa industrielle Kontroll- oder SCADA-Systeme, aber auch alle anderen vernetzten Geräte, die mechanische oder physikalische Auswirkungen haben können.

Red Team / Blue Team, rotes und blaues Team: Die roten Teams testen die Angriffswiderstandsfähigkeit der IT eines Unternehmens oder einer Organisation, indem sie versuchen, in die Systeme einzudringen und Angriffe zu simulieren. Die blauen Teams bestehen aus den Sicherheitsverantwortlichen einer Organisation, sie sollen die IT von innen schützen.

SCADA-Systeme, Supervisory Control and Data Acquisition: Systeme zum Überwachen und Steuern technischer Prozesse.

SIEM, Security Information and Event Management: SIEM-Systeme sammeln Daten aus Netzwerkkomponenten, Systemen und Anwendungen, werten sie aus und korrelieren sie, um sicherheitsrelevante Vorfälle frühzeitig zu erkennen.

SMB, Server Message Block: Netzwerkprotokoll für verschiedene Serverdienste in Rechnernetzen, etwa Datei- oder Druckdienste.

SOC, Security Operations Center: Organisationseinheit, die sich ausschließlich um die Cybersicherheit der IT-Systeme kümmert. Das schließt die permanente Überwachung der Sicherheit via Sensorik ein, aber auch das planvolle Reagieren bei Sicherheitsvorfällen.

SPS, speicherprogrammierbare Steuerungen: Gerät, das digital programmiert wird und der Steuerung einer Maschine oder Anlage dient. Gegensatz zur fest verdrahteten Steuerung.

TIBER-EU, Threat Intelligence-based Ethical Red Teaming: Rahmenwerk der Europäischen Zentralbank für kontrollierte Angriffe, um die Widerstandsfähigkeit des Finanzsektors gegenüber Cyberattacken zu testen.

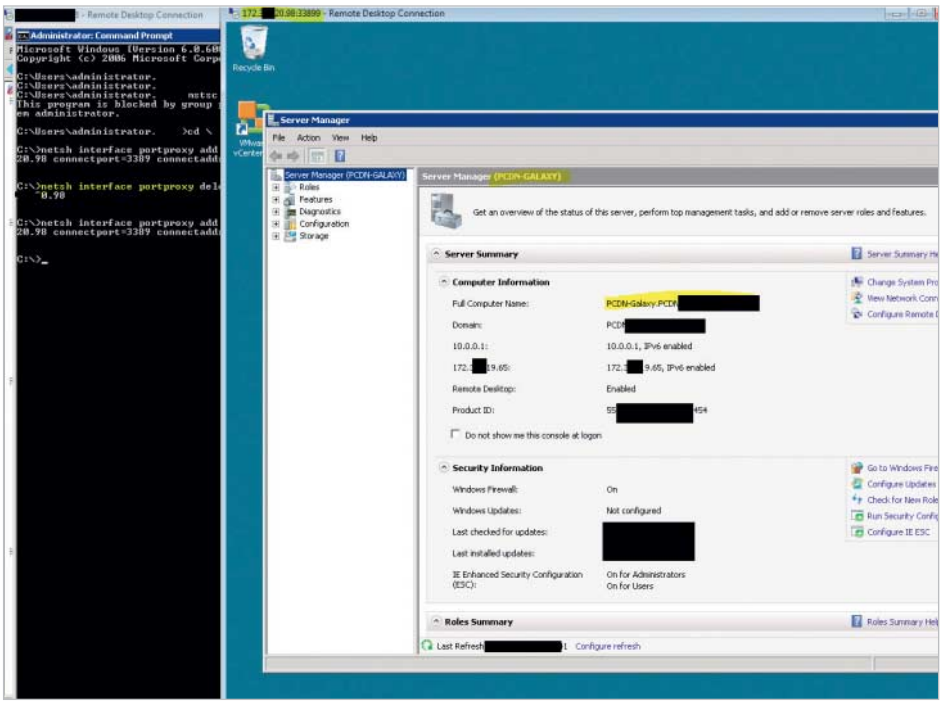
WMI, Windows Management Instrumentation: Microsofts Implementierung und Erweiterung des Standards Common Information Models (CIM) für das Management von IT-Systemen.

findet mit Domain-Admin-Rechten zuverlässig solche Systeme.

Über diesen Weg fanden wir das Manufacturing Execution System (MES), das schon per Definition einen Zugang in das Produktionsnetz benötigt. Dank der neu gewonnenen Domain-Admin-Rechte konnten wir uns auch an diesem System direkt über Remote Desktop anmelden und per Mimikatz die Passwörter des MES-Admin-Kontos aus dem Cache auslesen (Abbildung 4).

Auf dem System lief ebenfalls der Historian des SCADA-Systems, eine Software, die Datenbankanwendungen für operative Prozessdaten bereitstellt (siehe ix.de/ix1812082). Somit waren wir kurz vor dem Ziel. Es war dann möglich, über das dual-homed MES ein Port Forwarding auf beliebige Systeme innerhalb des Prozessleitnetzwerks zu schalten und das System als Proxy zu nutzen, wie im Listing „Port Forwarding" dargestellt.

Die Office-IT befand sich im Netzwerk 172.39.120.0/24, das Prozessleitnetzwerk im Segment 172.39.119.0/24. Nur das MES war in der Lage, beide Netzbereiche zu erreichen – was wir ausnutzen konnten, um Zugang auf die Steuerungsebene zu bekommen (Abbildung 5).



Über den Zugang zum MES war es letztlich möglich, bis zur Steuerungsebene zu gelangen (Abb. 5).

Am Ziel: Es kann manipuliert werden

Mit diesem Zugang wäre es einem Angreifer möglich, direkt auf das Human Machine Interface (HMI) zuzugreifen, Schwellenwerte der Sensorik direkt in der Datenbank zu manipulieren sowie direkten Zugriff auf die SPS-Systeme zu bekommen. So hätte er die Steuerung und Regelung der Anlage manipulieren können.

Da wir allerdings keine erfahrenen Ingenieure der Wasserwirtschaft sind, ließen wir die Finger von allen weiteren Aktionen, die vielleicht zu einer Kontamination von Trinkwasser hätten führen können, und begnügten uns mit dem Ergebnis. Wir konnten beweisen, dass es durch einen gezielten Angriff möglich ist, die Wasserfiltrierung zu kompromittieren. Außerdem konnten wir aufdecken, mit welchen TTPs (Tools, Techniques, Procedures) und aufgrund welcher fehlenden oder unzureichenden Security Controls ein solcher Angriff gelingen kann.

In einem abschließenden Workshop diskutierten wir mit Management, Projektleitern und Technikern, welche tatsächlichen Auswirkungen eintreten können, welche Maßnahmen sofort und mit wenig Aufwand umgesetzt werden sollten und wie Personal am besten ausgebildet und vorbereitet werden muss, um solche Angriffe erkennen und eindämmen zu können. Einige Maßnahmen finden sich im Kasten „Hilfreiche Tipps …".

Im nächsten Artikel der Serie geht es darum, wie man sich effektive Command-and-Control-Umgebungen aufbaut und sie managt, um unbemerkt persistenten Zugang in sein Zielnetzwerk zu erlangen. (ur@ix.de)

Hilfreiche Tipps und Schutzmaßnahmen

Hier eine (unvollständige) Liste mit einigen wichtigen Hinweisen, wie man sich vor unbefugtem Eindringen in die eigenen Systeme schützt:

- saubere interne Netzwerksegmentierung und -trennung durch Firewalls;
- generelle Reduzierung von Systemschnittstellen und Netzübergängen;
- Monitoring, Auditing und Logging aller internen Netzbereiche und Netzübergänge durch entsprechende Techniken (SIEM et cetera) und idealerweise Vorhandensein eines dedizierten SOC;
- regelmäßige Red-Team-Blue-Team-Übungen, um die Wirksamkeit der eingesetzten Maßnahmen zu prüfen;
- saubere Datenklassifizierung, um zu vermeiden, dass Unberechtigte Zugang zu technischen, systemrelevanten Daten innerhalb des internen Netzes erhalten;
- gut umgesetzte Incident-Response-Prozesse, um im Falle eines Angriffs effizient reagieren zu können (idealerweise im Zuge eines ISMS);
- Umsetzung einer sicheren Microsoft-Active-Directory-Umgebung (siehe dazu Microsofts Red-Forest-Umgebung, ix.de/ix1812082) sowie
- Reduzierung und sicherer Einsatz privilegierter Benutzerkonten, beispielsweise durch Microsofts LAPS (siehe ix.de/ix1812082).

Sascha Herzog

ist technischer Geschäftsführer und Penetrationstester bei der NSIDE ATTACK LOGIC GmbH in München.

Literatur

[1] Sascha Herzog; Awareness; Gesammeltes Wissen; Red Teaming: Taktische Informationsbeschaffung; *iX* 4/2018, S. 92

Alle Links: ix.de/ix1812082