Hacken met pdf-files

De risico's van kwaadaardige pdf-bestanden

Ir. M. Paques
is als adviseur werkzaam binnen de business unit IT Security en control van KPMG IT Advisory. Hij houdt zich onder meer bezig met security testing, social engineering, technische security reviews en de beveiliging van nieuwe technologieën.

paques.matthieu@kpmg.nl

Ir. Matthieu Paques

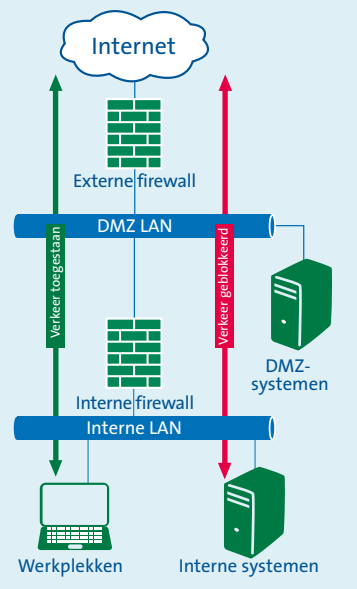
Waar een aanval van hackers zich in het verleden met name richtte op de kritieke serversystemen van een bedrijf, ontwikkelt zich een trend waarin aanvallen zich meer en meer ook richten op eindgebruikerssystemen ([Thomo1], [Micro8], [Govco8]). Het ontvangen van phishing mails en spyware is een toenemend verschijnsel ([Hillo4]). Niet alleen is er meer aandacht voor aanvallen op werkplekken, ook komen exploits[1], code die misbruik maakt van zwakheden in applicaties, in toenemende mate publiek beschikbaar. Een voorbeeld hiervan zijn de recente zwakheden in Adobe Reader. Met enige regelmaat worden nieuwe kwetsbaarheden in Adobe Reader vastgesteld en gepubliceerd. Maar hoe erg is dit nu eigenlijk? En in welke mate beschermen de aanwezige maatregelen als regelmatige updates, virusscanners en firewalls tegen deze bedreigingen? Wat bereikt een aanvaller met een dergelijke aanval en kan dit mogelijk impact hebben op de gewenste werkwijze tijdens een audit of accountantscontrole? In dit artikel zullen deze vragen beantwoord worden.

Inleiding

Al in 2005 werd een trend waargenomen van aanvallen waarbij de focus op eindgebruikers lag ([Millo5]). Een aspect hiervan is het snel in aantal toenemen van het aantal spyware- en phishingincidenten ([Hillo4]). De aanvallen verschuiven hierbij volgens CERT ([Millo5]) veelal van een IT-infrastructuurgerichte aanval naar een applicatiegerichte aanval. In een dergelijke aanval stuurt een aanvaller een gebruiker een geïnfecteerd bestand dat vervolgens door de gebruiker in een kwetsbare applicatie wordt geopend. Bij openen van de file wordt vervolgens een door de aanvaller geprogrammeerde actie uitgevoerd. Vanuit het perspectief van een aanvaller zijn er legio mogelijkheden om een geïnfecteerd bestand bij gebruikers te krijgen, denk aan:

- webbrowsers (forums, gratis downloads);
- instant messenger clients;
- e-mail (phishing mails) met een vervalst (gespoofed) afzenderadres. In geval van een bedrijf kan hier mogelijk een andere medewerker van het bedrijf als afzender worden gebruikt, denk aan een 'belangrijke mededeling van de raad van bestuur';

1 Code die misbruik maakt van een kwetsbaarheid in bepaalde software, veelal met het doel om toegang te krijgen tot een systeem, verkrijgen van meer rechten of uitvoeren van een zogenaamde Denial-of-Service-aanval. Het Metasploit framework is een bekende tool voor het ontwikkelen en uitvoeren van exploits.



Figuur 1. **Verkeer door de firewall.**

- peer-to-peer netwerken of torrents;
- andere gebruikersapplicaties ([Cisco8]).

Een trend die zich naar verwachting in 2010 verder zal doorzetten, is de aanval via social networking sites als Myspace, Facebook, Hyves en andere ([Malw09]).

Eindgebruikerssystemen hebben veelal toegang tot het internet. Met andere woorden, het is mogelijk een verbinding op te zetten door de firewall heen vanaf deze systemen, bijvoorbeeld met een webbrowser. Serversystemen daarentegen zijn veelal niet direct vanaf het internet bereikbaar. Verkeer dat servers vanaf het internet probeert te benaderen wordt dan door een firewall geblokkeerd (zie figuur 1).

Wanneer een eindgebruikerssysteem wordt gecompromitteerd (overgenomen door een hacker) kan dit worden gebruikt als stepping-stone (uitgangspunt) voor verdere aanvallen op het netwerk. Om kritieke serversystemen te bereiken voert een hacker een dergelijke aanval in twee stappen uit. In de eerste stap wordt een eindgebruikerssysteem overgenomen, waarna vanaf het overgenomen systeem – zonder gehinderd te worden door maatregelen als de externe firewall – een aanval op kritieke systemen wordt uitgevoerd. Hiernaast is een aanvaller in staat data van het overgenomen eindgebruikerssysteem te kopiëren of aan te passen of het systeem op te nemen in een zogenaamd 'botnet'.[2]

2 Een leger van computers die, vaak zonder dat de eigenaars dit weten, worden ingezet voor Denial-of-Service (DoS)-aanvallen.

Aanvallen langs vele wegen

Aanvallen op gebruikerssystemen door middel van kwetsbaarheden in applicaties

Een gemiddelde gebruiker heeft bij benadering zo'n tachtig applicaties geïnstalleerd staan ([Ballo8], [Ballo9]). Uit statistieken blijkt dat gemiddeld 85 procent van deze applicaties is gepatcht tegen securitybedreigingen, wat neerkomt op gemiddeld twaalf niet-gepatchte applicaties per gebruiker (zowel wereldwijd als in Nederland). In 2008 bleek uit onderzoek op 20.000 gebruikerssystemen dan 89,09 procent hiervan minimaal één 'onveilige' applicatie bevatte.

Number of insecure programs per PC/user:	
0 Insecure Programs:	1.91% of PCs
1-5 Insecure Programs:	30.27% of PCs
6-10 Insecure Programs:	25.07% of PCs
11+ Insecure Programs:	45.76% of PCs

Bron: [Ballo8]

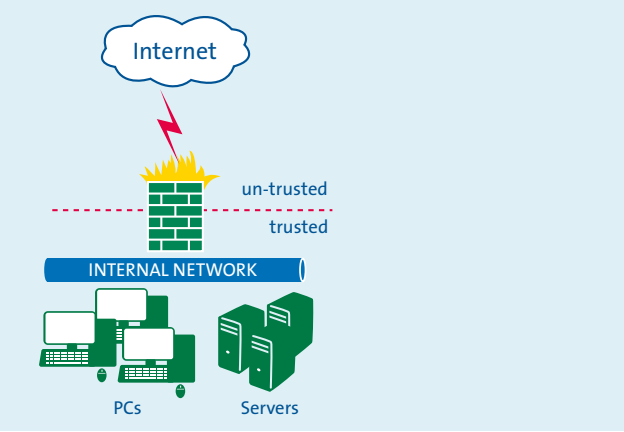
Met 'onveilige' applicatie wordt hier een applicatie bedoeld waarvan de producent een nieuwere versie heeft uitgegeven waarin één of meer kwetsbaarheden zijn opgelost, maar de gebruiker deze nog niet heeft geïnstalleerd. Merk op dat geïdentificeerde kwetsbaarheden waar nog geen patch voor beschikbaar is, maar mogelijk wel exploits voor bestaan, niet in deze statistieken zijn meegenomen.

Veelgebruikte applicaties op eindgebruikerssystemen waar virussen en exploits misbruik van maken zijn onder andere:

- Word, PowerPoint, Excel;
- Adobe Reader, Flash Player;
- Mozilla Firefox, Internet Explorer.

Omzeilen van firewalls

Beschermende maatregelen binnen een organisatie zijn in de praktijk veelal gericht tegen aanvallen van buitenaf. In een externe firewall kan daartoe al het verkeer dat van buiten een verbinding tot stand probeert te brengen met een intern systeem, worden geblokkeerd. Vaak is het echter wel mogelijk om vanuit een 'vertrouwd' intern systeem een verbinding met een extern systeem op te zetten en daarmee verkeer door de firewall te laten sturen. Juist dit laatste kan in een aanval gebruikt worden door een gebruiker een geïnfecteerd pdf-bestand te sturen waarbij *vanaf* de werkplek een verbinding wordt opgezet naar buiten, waar de aanvaller zich bevindt.



Figuur 2. **Maatregelen tegen aanvallen van buitenaf. (Bron: http://www.networksurety.com/solutions/firewall.php.)**

Aanvallen vanuit het interne netwerk

Wanneer een penetratietest vanuit het *interne* netwerk wordt uitgevoerd (bijvoorbeeld op serversystemen die relevant zijn voor de financiële verslaggeving), blijkt dat in ruim 90 procent[3] van de gevallen dat kwetsbaarheden aanwezig zijn die vanuit het gebruikerssegment misbruikt kunnen worden. Dit in tegenstelling tot externe penetratietesten vanaf het internet waar dit percentage aanvallen, waarbij daadwerkelijk de achterliggende systemen geraakt worden, beduidend lager ligt. Wanneer een aanvaller van buitenaf dus in staat is – bijvoorbeeld door middel van een pdf-exploit – een gebruikerssysteem op het interne netwerk te compromitteren, kan hiervandaan een verdere aanval op het interne netwerk worden uitgevoerd.

Kwetsbaarheden op eindgebruikerssystemen kunnen aldus indirect leiden tot het compromitteren van voor de jaarrekeningcontrole relevante systemen.

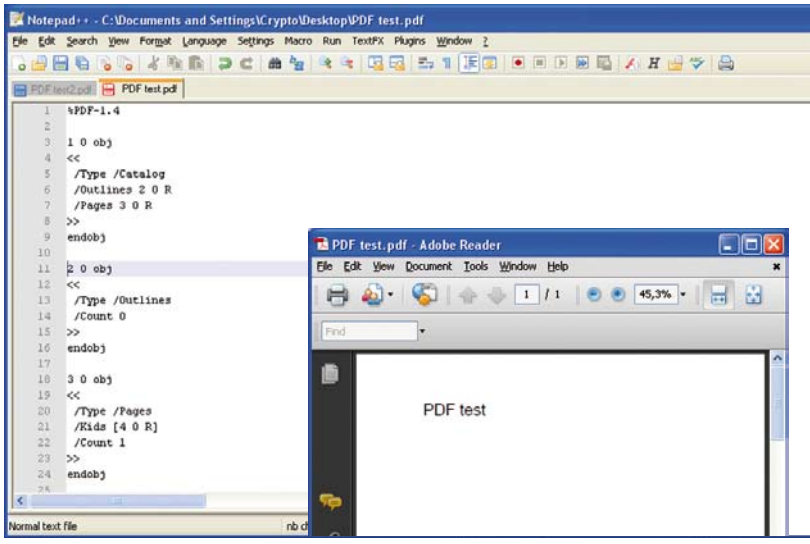
Aanvallen door middel van pdf-bestanden

Recente versies van Adobe Reader bevatten kwetsbaarheden die kunnen leiden tot volledige (externe) toegang tot het systeem ([Goodo7], [Lauo7], [MacAo8], [Nazao8], [Truso8], [Millo5]). Zelfs wanneer een gebruiker Adobe Reader direct vanaf de officiële site van Adobe downloadt, bestaat de kans dat dit een kwetsbare versie betreft. Op het moment van schrijven (8 augustus 2009) wordt bij installatie vanaf de Adobe website (http://get.adobe.com/reader/) versie 9.1.0 geïnstalleerd, als standalone programma en als browser-plugin. Deze versie staat bekend als kwetsbaar voor verschillende zogenaamde 'code execution' kwetsbaarheden. De patches die in mei 2009 (één kwetsbaarheid) en juni (negen kwetsbaarheden) zijn uitgegeven, zijn niet aan de aangeboden versie toegevoegd. Het wordt aan de gebruiker overgelaten om de zojuist gedownloade versie direct weer te updaten.

3 Dit percentage is gebaseerd op de resultaten van een groot aantal interne penetratietesten uitgevoerd door KPMG.

Analyse van een pdf-bestand

Een pdf-document bestaat uit alleen ASCII-karakters en kan daarom met een eenvoudige editor als Notepad worden gelezen ([Stevo9]). Hiernaast bevat een pdf-document veel whitespaces en wordt er geen gebruik gemaakt van compressie, wat de leesbaarheid vergroot. In figuur 3 wordt de code van een eenvoudige pdf-file getoond die alleen de tekst 'PDF test' bevat.



Figuur 3. **Opbouw van een pdf-bestand.**

Hieronder worden de basiscomponenten van een eenvoudige pdf beschreven. Voor meer gedetailleerde informatie, zie [PDFD].

Header

In de header wordt de pdf-versie gespecificeerd.

```
%PDF-1.4
```

Objecten

Objecten bepalen de structuur en inhoud van het pdf-document. Objecten beginnen met een uniek referentienummer gevolgd door een versienummer. In het voorbeeld pdf-document wordt een zestal objecten gedefinieerd. In het eerste plaatje wordt een object gedefinieerd met referentienummer 1 en versienummer 0. Tevens wordt het type (catalog) aangegeven. De regels 4 en 5 in dit voorbeeld bevatten een referentie naar indirecte objecten 2 en 3 die later in de code beschreven zijn.

```
1 0 obj
<<
 /Type /Catalog
 /Outlines 2 0 R
 /Pages 3 0 R
>>
endobj
```

Object 2 beschrijft de outline voor het document. Die is in dit geval gelijk aan 0. De pagina's in het document worden beschreven in object 3. Het /Kids-element verwijst hierin naar de inhoud van de pagina's. Het /Count-element definieert het aantal pagina's (1) van het document.

```
2 0 obj
<<
 /Type /Outlines
 /Count 0
>>
endobj
```

```
3 0 obj
<<
 /Type /Pages
 /Kids [4 0 R]
 /Count 1
>>
Endobj
```

Object 3

Object 4 beschrijft de afmeting van de pagina (Mediabox) en bevat referenties naar de inhoud van de pagina (object 6) en het gebruikte font in de pagina (object 5). De definitie van het font is weergegeven in deze afbeelding.

```
4 0 obj
<<
 /Type /Page
 /Parent 3 0 R
 /MediaBox [0 0 612  792]
 /Contents 6 0 R
 /Resources <<
        /ProcSet [/PDF /Text]
        /Font << /F1 5 0 R >>
       >>
>>
Endobj
```

```
5 0 obj
<<
 /Type /Font
 /Subtype /Type1
 /Name /F1
 /BaseFont /Helvetica
 /Encoding /MacRomanEncoding
>>
Endobj
```

De inhoud van de pagina (object 6) wordt beschreven in een zogenaamde 'stream'. Een stream kan verschillende encodings bevatten. In dit geval bevat de stream de tekst 'PDF test' in font F1 en lettergrootte 18. De tekst wordt geprint op locatie 100.700 van de pagina.

```
6 0 obj
<< /Length 50 >>
stream
BT /F1 18 Tf 100 700 Td (PDF test) Tj ET
endstream
endobj
```

Xref en trailer

Tot slot worden de Xrefs gedefinieerd. De Xrefs zijn een index voor de pdf-applicatie en bevatten de absolute offsets (verwijzing) naar de aanwezige objecten (het aantal karakters vanaf het begin van de file tot aan het object). Deze index begint met een offset naar het zogenaamde '0-object' gevolgd door de offsets naar de andere zes objecten in het document. Het eerste object heeft dus een offset van 12, met andere woorden object 1 start 12 bytes na het begin van de file. De trailer geeft aan met welk object de pdf-applicatie dient te beginnen met lezen, in dit geval dus object 1. De startxref bevat een verwijzing (offset) naar de absolute locatie van de xrefs in de file.

```
xref
0 7
0000000000 65535 f
0000000012 00000 n
0000000089 00000 n
0000000145 00000 n
0000000214 00000 n
0000000419 00000 n
0000000520 00000 n
trailer
<<
 /Size 7
 /Root 1 0 R
>>
startxref
645
%%EOF
```

Toevoegen van JavaScript in pdf

Het pdf-formaat ondersteunt interactieve content als JavaScript. Ingevoegd JavaScript kan bijvoorbeeld worden uitgevoerd wanneer het document wordt geopend (via /Openaction) of wanneer een specifieke pagina in het document wordt benaderd. In de bovenstaande voorbeeld-pdf kunnen we JavaScript laten uitvoeren bij het openen van het document door het toevoegen van het volgende object (object 7):

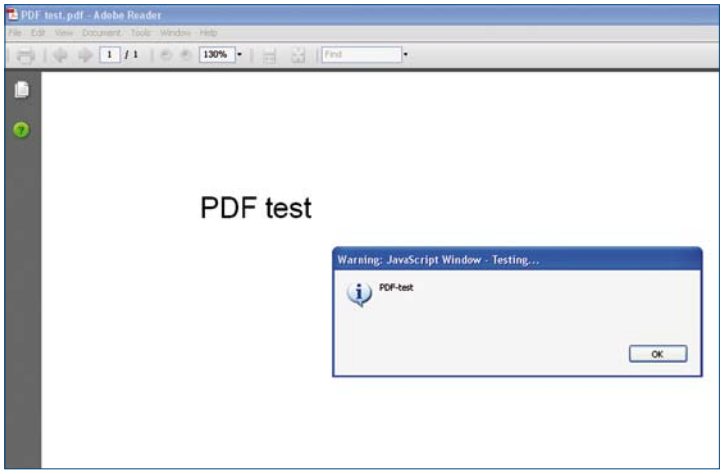
```
7 0 obj
<<
 /Type /Action
 /S /JavaScript
 /JS (app.alert({cMsg: 'PDF-test', cTitle: 'Testing...', nIcon: 3});)
>>
Endobj
```

Tevens voegen we vanuit het root object een referentie toe (/OpenAction), zodat het object wordt gestart wanneer de file wordt geopend:

```
1 0 obj
<<
 /Type /Catalog
 /Outlines 2 0 R
 /Pages 3 0 R
 /OpenAction 7 0 R
>>
Endobj
```

Bij het openen van het document wordt het script uitgevoerd en (in dit geval) een pop-up getoond (zie figuur 4).

De JavaScript engine voor pdf is erg beperkt. Echter, door gebruik te maken van kwetsbaarheden in Adobe Reader is het mogelijk meer functionaliteit toe te voegen.



Figuur 4. **JavaScript uitvoeren vanuit een pdf-file.**

Misbruik van kwetsbaarheden in Adobe Reader

In Adobe Reader worden in toenemende mate kwetsbaarheden gemeld. Een voorbeeld is de *Util.printf* kwetsbaarheid die in het kader hiernaast in meer detail wordt toegelicht. Enkele andere kwetsbaarheden waar kant-en-klare exploits voor beschikbaar zijn, zijn weergegeven in figuur 5.

```
>> use exploit/windows/fileformat/adobe_collectemailinfo
>> use exploit/windows/fileformat/adobe_geticon
>> use exploit/windows/fileformat/adobe_jbig2decode
>> use exploit/windows/fileformat/adobe_pdf_embedded_exe
>> use exploit/windows/fileformat/adobe_utilprintf
```

Figuur 5. **Beschikbare pdf-exploits.**

Tools voor het automatisch creëren van kwaadaardige pdf-bestanden

Online zijn verschillende zogenaamde 'toolkits' beschikbaar die een grote hoeveelheid exploitcode bevatten. Met een dergelijke toolkit kan op eenvoudige wijze een geïnfecteerd bestand worden gemaakt. Eén van de bekendere toolkits is het Metasploit framework. De *Util.printf* exploit is inmiddels als Mixin (plugin) voor het Metasploit framework beschikbaar. Via Metasploit kan eenvoudig de *Util.printf()* payload[4] worden geselecteerd die deze vervolgens automatisch in een pdf injecteert. Metasploit is vrij online beschikbaar.

Util.printf() buffer overflow (Adobe Reader 8.1.2)

Een van de kwetsbaarheden waar eind vorig jaar veel exploits misbruik van maakten, betreft een buffer overflow[5] in Adobe Reader. Deze kwetsbaarheid wordt veroorzaakt door een fout in de util.printf() JavaScript-functie. Om deze kwetsbaarheid te misbruiken dient een aanvaller een hiertoe geprepareerde pdf-file door een reguliere gebruiker op een doelsysteem te laten openen. Wanneer de pdf-file wordt geopend, wordt de hierin door de aanvaller verborgen code uitgevoerd onder de rechten van de reguliere gebruiker.
Onderstaande Python-code (volledige toelichting op: http://www.coresecurity.com/content/adobe-reader-buffer-overflow) maakt gebruik van deze buffer overflow. Deze code genereert JavaScript-code die, wanneer uitgevoerd vanuit een pdf, probeert code op geheugenadres 0x30303030 uit te voeren. Wanneer we een string toewijzen in JavaScript wordt deze op een willekeurige plaats in het geheugen geschreven. Om shellcode op adres 0x30303030 te krijgen kan een groot aantal strings naar het geheugen worden geschreven om daarmee de kans te vergroten dat één van deze strings op adres 0x30303030 wordt geplaatst. Deze techniek wordt ook wel heap-spraying genoemd.

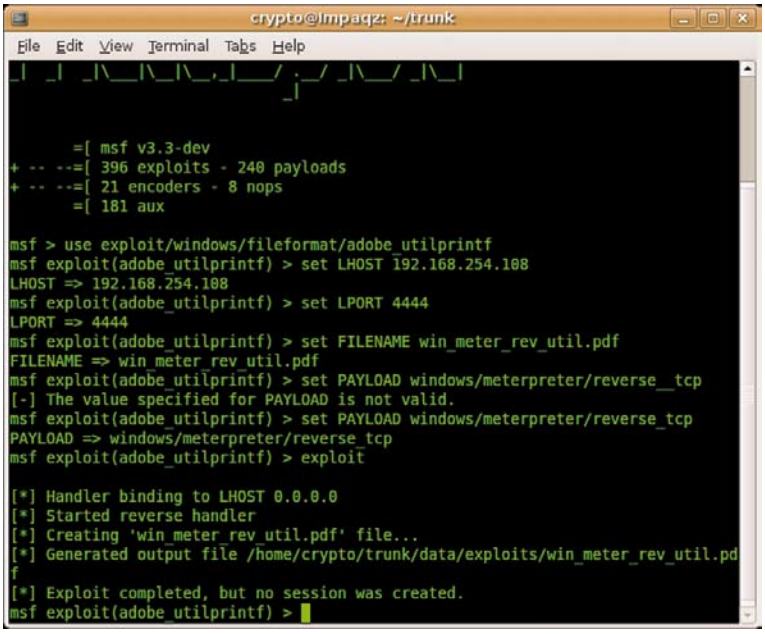
```
fill = 276 * '8'
    script="""
    var num = 12999999999999999999_FILL_
    util.printf("%45000f",num)
    """
    script = script.replace('_FILL_', fill)

    print script
```

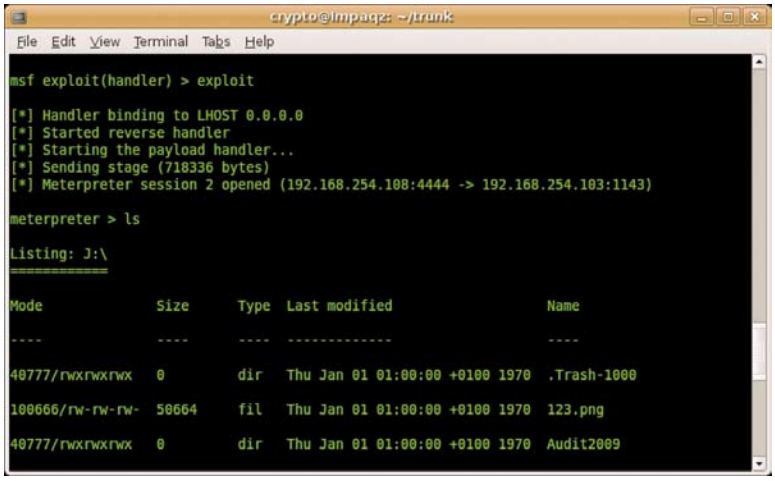
De shellcode dient gestart te worden vanaf de eerste instructie. In het ideale geval zou de eerste instructie van het programma dus precies op geheugenadres 0x30303030 terechtkomen. Omdat niet vastgesteld kan worden op welk punt van de stack de huidige code zich bevindt en na hoeveel strings geheugenadres 0x30303030 is bereikt, laten we de shelllcode voorafgaan door een lange zogenaamde NOP-sled.
Een NOP-instructie (No OPeration) bestaat uit slechts één byte en voert geen actie uit, maar vervolgt met de volgende NOP-instructie. Op deze manier kan een groot aantal NOP-instructies doorlopen worden totdat de shellcode bereikt en uitgevoerd wordt. Voor de shellcode maakt het niet uit met welke NOP-instructie er gestart wordt. De hierop gebaseerde exploit ([Millo5]) is op deze wijze opgebouwd. De payload bevat de uit te voeren code.

4 Shellcode (ook wel payload): code die wordt uitgevoerd door de exploit. Bevat de opdracht die op de doelmachine wordt uitgevoerd, veelal het opstarten van een command Shell.

5 Een buffer overflow is een foutief gedrag van software die probeert data te schrijven in een tijdelijke gegevensruimte maar buiten de grenzen van deze buffer schrijft. Dit veroorzaakt in de meeste gevallen een stopzetting van het programma. Fouten in applicaties, maar nog meer in het geval van besturingssystemen, worden soms misbruikt door wormen, virussen en hackers om ongeoorloofde toegang te krijgen tot computersystemen.



Figuur 6. **Het creëren van een kwaadaardige pdf.**



Figuur 7. **Opvragen van op het gecompromitteerde systeem actieve processen.**

In figuur 6 wordt weergegeven hoe met behulp van Metasploit een pdf kan worden gecreëerd die geïnfecteerd is met de genoemde *Util.printf() exploit*.

Nadat de gebruiker de pdf heeft geopend, wordt automatisch een verbinding gemaakt met de aanvaller en een zogenaamde 'shell' gecreëerd. De aanvaller kan vervolgens het gecompromitteerde systeem bedienen alsof hij fysiek op het systeem werkt met de rechten van de ingelogde gebruiker. In figuur 7 is te zien hoe de aanvaller een overzicht opvraagt van de op het gecompromitteerde systeem actieve processen. Andere mogelijkheden zijn het bekijken of wijzigen van alle gegevens op het systeem zoals alle wachtwoordhashes van gebruikers.

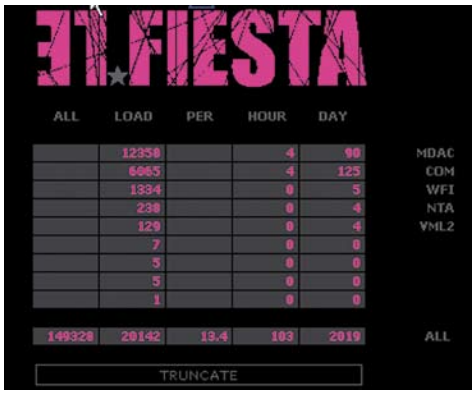
Een alternatief voor het creëren van een shell is het injecteren van een VNC-server. Hiermee kan de aanvaller grafisch meekijken op het scherm van het geïnfecteerde systeem en desgewenst de (muis en toetsenbord) besturing overnemen. Een voorbeeld hiervan is te zien in figuur 8. De aanvaller kan hier precies zien wat er op de desktop van de gebruiker gebeurt en 'live' op het scherm meekijken.

Geautomatiseerd verspreiden van malware (onder andere pdf)

Verspreiden van malware kan, zoals eerder opgemerkt, op verschillende manieren plaatsvinden: via e-mail, peer-to-peer netwerken of websites. Voor verspreiding van exploits via het web zijn op de zwarte markt verschillende zogenaamde 'exploit packs' te verkrijgen. Eén van deze exploit packs is *El Fiesta*, dat voor rond de $ 800 wordt aangeboden op Russische fora. Dit exploits pack bevat een vijfentwintigtal exploits (afhankelijk van de versie), waaronder de eerder beschreven pdf-exploit.

Een dergelijk exploit pack bevat hiernaast over het algemeen een webpagina die de aanvaller online kan plaatsen. Wanneer deze wordt bezocht door een bezoeker doorloopt het exploit pack alle aanwezige exploits om vast te stellen of het systeem van de bezoeker kwetsbaar is voor één van deze. Wanneer het systeem van de bezoeker kwetsbaar is wordt dit automatisch geïnfecteerd met een van tevoren bepaalde payload en worden zijn gegevens (IP-adres) naar een database geschreven. De beheerder van de exploitpagina kan rustig afwachten en in een overzicht zien hoeveel gebruikers reeds geïnfecteerd zijn en – in geval van een botnet – de geïnfecteerde clients (bots) opdrachten geven.

Figuur 8. **Overnemen van het scherm door middel van VNC.**

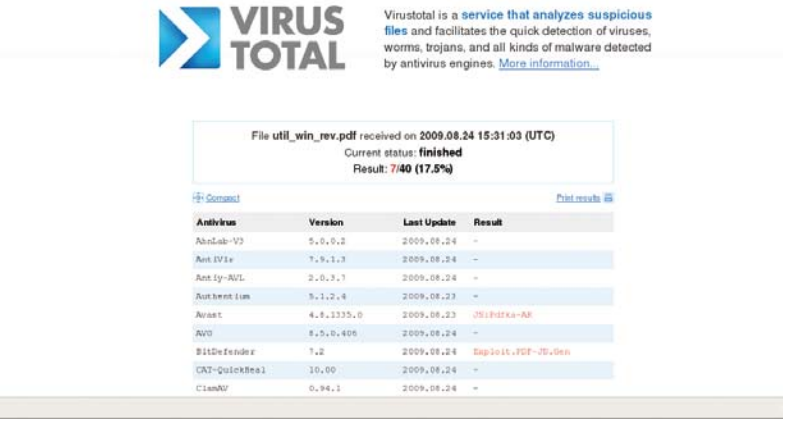


Figuur 9. **Overzicht van geïnfecteerde systemen in El Fiesta.**

Beheersingsmaatregelen en mogelijke gevolgen voor de financiële controle

De hierboven geproduceerde pdf is getest op een systeem met Windows XP SP2 en verschillende Adobe Reader-versies. De pdf kan op alle kwetsbare pdf-versies succesvol worden uitgevoerd.

Het merendeel van de virusscanners herkent de geproduceerde pdf-file nog niet als kwaadaardig. In figuur 10 staan de resultaten van Virustotal, waar het bestand door veertig verschillende virusscanners is onderzocht. Slechts zeven van deze scanners gaven aan dat het bestand kwaadaardige code bevatte.



Figuur 10. **Resultaten van Virustotal.**

Naast het gebruik van een virusscanner is het mogelijk om handmatig een pdf te bekijken met een teksteditor. Verdachte strings zijn /JS en /JavaScript die een indicatie zijn voor JavaScript in de pdf-file. Praktisch alle kwaadaardige pdf-bestanden die ik ben tegengekomen bevatten JavaScript. Vanzelfsprekend betekent het aantreffen van JavaScript niet per definitie dat het een kwaadaardig pdf-bestand betreft.

Naast het bijwerken van antivirusdefinities zijn er verschillende aanvullende maatregelen die genomen kunnen worden om risico's van applicatiegerichte aanvallen tegen te gaan, waaronder:

- het beperken van rechten van lokale gebruikers. Lokale gebruikers dienen op basis van het 'need to have'-principe ingericht te zijn, en niet de mogelijkheid te hebben zelf applicaties te installeren;
- het filteren van outboundconnecties in de firewall om ongewenste verbindingen naar buiten tegen te gaan;
- het inrichten van adequate patch-managementprocessen, voor zowel besturingssystemen als applicaties. Er zijn verschillende applicaties die checken of de meest recente versies van applicaties geïnstalleerd zijn, zoals Updatestar, Updatechecker en de PSI-scanner van Secunia;
- het inrichten van monitoring om afwijkingen van het toegangsbeleid te detecteren en de te controleren gebeurtenissen te registreren, als bewijs bij beveiligingsincidenten. Systeemmonitoring maakt het mogelijk zeker te stellen dat gebruikers uitsluitend processen uitvoeren waarvoor zij expliciet zijn gemachtigd;
- in het incident-managementproces aandacht besteden aan de opvolging van (mogelijk) geïnfecteerde eindgebruikerssystemen;
- het inrichten van adequate netwerkscheiding tussen eindgebruikerssystemen en servers (gelaagd netwerkbeveiligingsmodel);
- aandacht besteden aan security (awareness) training. Gebruikers dienen te worden getraind in het omgaan met de beveiligingsprocedures en het correcte gebruik van IT-voorzieningen, om eventuele beveiligingsrisico's te minimaliseren.

Een accountant kan vaststellen in welke mate deze maatregelen zijn ingericht om de noodzaak van het controleren van de veiligheid van werkplekken op te nemen in het controleprogramma.

Conclusie

Sinds een aantal jaren vindt een ontwikkeling plaats waarbij aanvallen zich in toenemende mate op gebruikerssystemen richten. Gebruikers worden hiertoe via allerlei verschillende kanalen benaderd.

Wanneer op werkplekken kwetsbare applicaties aanwezig zijn, zoals een kwetsbare versie van Adobe Reader, bestaat de kans dat een aanvaller die hier misbruik van maakt volledige controle krijgt over de betreffende desktop. Indien een aanvaller in staat is een werkpleksysteem op het interne netwerk te compromitteren (bijvoorbeeld door kwetsbaarheden in het besturingssysteem of applicaties) vergroot deze hiermee de mogelijkheden om een server die relevant is voor de jaarrekeningcontrole tevens te compromitteren aanzienlijk. Dit kan grote impact hebben op

de integriteit van daarop aanwezige financiële gegevens en daaraan gerelateerde controles. In een veelvoud van praktijksituaties van de auteur is gebleken dat toegang tot een eindgebruikerssysteem voldoende is om serversystemen te compromitteren en de financiële verslaggeving of andere data in te zien dan wel aan te passen.

Maatregelen als firewalls, virusscanner en updates van besturingssystemen hebben slechts beperkte invloed op het tegengaan van deze bedreiging. Hoewel voor kwetsbaarheden in de applicaties regelmatig patches uitkomen, blijven er ook nieuwe kwetsbaarheden ontdekt worden.

Om vast te stellen of werkplekken gedurende een onderzoeksperiode kwetsbaar zijn geweest, is het van belang de aanwezigheid van beperkende maatregelen zoals deze in voorgaande paragraaf zijn toegelicht, gedurende deze periode te kunnen vaststellen.

Literatuur

[Ball08] Jakob Balle, *1.91% of all PCs are fully patched!*, http://secunia.com/blog/37/ 3rd December 2008

[Ball09] Jakob Balle, http://secunia.com/blog/56/, 25th June 2009

[Cisco08] *Targeted Internet attacks will increase in 2009*, December 19, 2008, Cisco http://www.internet-security.ca/internet-security-news-020/cisco-report-targeted-internet-attacks-will-increase-in-2009.html

[Fies] *Fiesta 2.4 – Monitoring ITW exploit*, http://www.prevx.com/blog/107/Fiesta---Monitoring-ITW-exploit.html

[Good07] D. Goodin, *Nasty PDF exploit runs wild*, October 2007, http://www.theregister.co.uk/2007/10/24/pdf_exploit_in_the_wild/

[Govc08] Govcert , juli 2008, *Trendrapport 2008: cybercrime in trends en cijfers*, http://itknowledgebase.computable.nl/rapport-detailpagina.183112.lynkx?rapportPointer=9-212545-212547-212551&filterValue=tactieken&filterType=&pageStart=

[Hill04] Gijs Hillenius, 25 maart 2004, *Code van virussen, spam en spyware versmelt*, http://www.computable.nl/artikel/ict_topics/security/258697/1276896/code-van-virussen-spam-en-spyware-versmelt.html

[Lau07] H. Lau, *When PDF's Attack... Again!*, October 2007, https://forums.symantec.com/t5/Vulnerabilities-Exploits/When-PDF-s-Attack-Again/ba-p/305545#A125

[MacA08] MacAfee, *Exploit-PDF.a*, October 2008, http://vil.nai.com/vil/content/v_139103.htm

[Malw09] http://www.h-desk.com/articles/Malware_Trends__What_Will_Attack_Us_in_2009__a45_f1.html

[Micr08] *Microsoft Security Intelligence Report volume 6* (July–December 2008), http://www.microsoft.com/downloads/details.aspx?FamilyID=aa6e0660-dc24-4930-affd-e33572ccb91f&displaylang=en

[Mill05] Jason Milletary, October 4, 2005, *Trends in Internet Attack, Technology and the Role of Artifact Analysis*, http://www.arcert.gob.ar/tc/presentaciones/martes/TendenciasEnCodigoMalicioco-CERTcc.pdf

[Milw] Util.printf exploit http://www.milworm.com/exploits/7006

[Naza08] J. Nazario, *PDF Exploit – In the wild, and how to decode*, November 2008, http://asert.arbornetworks.com/2008/11/pdf-exploit-in-the-wild-and-how-to-decode/

[PDFD] PDF Dictionary, http://1t3xt.info/api//com/lowagie/text/pdf/PdfDictionary.html

[Stev09] Didier Stevens, Haking 3/2009, *Anatomy of malicious PDF Documents*

[Thom01] Ian Thomson, *Users left open to attack by failure to patch third-party apps*, 21 Apr 2009, http://www.v3.co.uk/vnunet/news/2240702/users-patching-third-party-apps

[Trus08] Trustedsource Anti-Malware Team, *Rise Of The PDF Exploits*, September 2008, http://www.trustedsource.org/blog/153/Rise-Of-The-PDF-Exploits

Tools

Updatestar: http://www.updatestar.com/

Metasploit 3.3: http://www.metasploit.com/framework/download/

Adobe Acrobat Reader 7, 8.1.1, 8.1.2, 9: http://www.oldversion.com/download_Acrobat_Reader_7.05.html

Updatechecker: http://www.filehippo.com/updatechecker/

Secunia: http://secunia.com/vulnerability_scanning/