**Sebastian Nemetz**

# Überblick über staatliche Spähsoftware

Stuxnet, Red October, Flame – *von Staaten entwickelte Schadsoftware, auch* Milware *genannt, unterscheidet sich in Aufbau und Funktionalität von nicht staatlicher Malware. Dieser Beitrag beleuchtet Charakteristiken von Milware, anwendbare Analysemethoden sowie Konsequenzen des Einsatzes von Milware und gibt anhand von sieben konkreten Beispielen einen fundierten Überblick über deren Verbreitungsweise, Exploits, technische Funktionsweise und tatsächlich bisher erreichte Fähigkeiten. Der Fokus liegt auf Milware, die durch akademische oder andere zuverlässige Quellen analysiert und dokumentiert wurde. Es werden auch Einsatzzwecke, Urheber und Opfer der jeweiligen Milware behandelt.*



In den letzten zehn Jahren waren Regierungen und Geheimdienste für Entwicklung und Verbreitung vieler hochkomplexer Schadprogramme (Malware) verantwortlich. Nicht nur deren Einsatz ist eine neue Herausforderung für den Bereich der Informationssicherheit, staatliche Schadsoftware stellt auch eine neue Dimension von Malware hinsichtlich ihrer Komplexität und Zielsetzung dar. Sie unterscheidet sich in vielen Punkten von herkömmlicher, nicht staatlicher Malware, weswegen für diese Kategorie der Begriff *Milware* eingeführt wurde[1]. Aufgrund des rechtlichen Status von Regierungen und der zur Verfügung stehenden finanziellen und personellen Ressourcen haben Staaten gegenüber nicht staatlichen Organisationen im Hinblick auf Innovationskraft, Vielfalt und Umfang einen enormen Vorteil bei Entwicklung und Einsatz ihrer Schadsoftware, weswegen Milware auch durch deutlich höheren Entwicklungsaufwand gekennzeichnet ist. Die Einführung des Begriffs vereinfacht eine Kategorisierung der von Sicherheitsfirmen gefundenen Programme zum Zwecke der Spionage (mitunter auch zur Zerstörung), deren Urheber Geheimdienste oder militärische Organisationen sind. Diese Programme werden auch dann als Milware bezeichnet, wenn sie zwar staatlich in Auftrag gegeben und genutzt werden, die Entwicklung aber an private Unternehmen ausgelagert worden ist.[2]

Der Begriff *Cyberwaffe* (*cyber weapon*) wird fälschlicherweise häufig synonym zu Milware gebraucht. Als Cyberwaffen werden nur die Schadprogramme bezeichnet, welche darauf ausgelegt sind, physischen oder logischen Schaden anzurichten. Während etwa Stuxnet eindeutig dazu zählt, gehört die meiste Milware/Malware nicht zu den Cyberwaffen, da sie zur Spionage und für Informationsdiebstahl entworfen wurde[2].

Um Milware ranken sich viele Mythen, staatlicher Schadsoftware werden oftmals enorme Fähigkeiten zugesprochen. In nachfolgenden Analysen wird dargelegt, wie die Quellenlage allgemein für die jeweilige Milware ist, welche Erkenntnisse durch akademische Quellen gesichert und zu welchen Ergebnissen andere Untersuchungen gekommen sind.

## Unterschiede zwischen Malware und Milware

Malware versucht nach dem Gießkannenprinzip, eine möglichst große Verbreitung zu erlangen, um möglichst oft ihren Schadcode ausführen zu können.[1] Milware hingegen hat meist eine kleine Zielgruppe und sucht in erster Linie Zugriff auf spezielle Rechner, um zu einem späteren Zeitpunkt ihre Wirkung entfalten zu können. Auch bei Datendiebstahl und anderen nicht zerstörerischen Funktionen ist die Zielsetzung bei Milware eine nicht kommerzielle, während Malware häufig finanziellen Interessen dient.


*Christoph Scholz, CC BY-SA 2.0*

Bisher versuchte man die Unterschiede zwischen staatlichen und nicht staatlichen Programmen durch Begriffe wie *Codekomplexität* auszudrücken. In einem neuen Ansatz generiert Trey Herr mit Hilfe einiger Metriken einen *MAlicious Software Sophistication Index*, kurz *MASS Index*, mit dem man Code als staatlich oder nicht staatlich klassifizieren können soll.[2] Dieser orientiert sich an funktionalen Gesichtspunkten und es werden High-Level-Charakteristiken wie Architektur und Verhaltensmerkmale der Schadsoftware zu Rate gezogen. Der *MASS Index* ist allerdings nicht als quantifizierendes Tool gedacht, sondern hat das Ziel, von qualitativer und deskriptiver Natur zu sein.

Ferner gilt es natürlich zu beachten, dass nicht nur zwischen staatlicher und nicht staatlicher Software eine Trennlinie verläuft, sondern auch zwischen Staaten: Werkzeuge aus Italien, Pakistan, Nordkorea, den USA oder China unterscheiden sich erheblich in ihrer Qualität. Selbst zwischen verschiedenen Organisationen eines Staates gibt es Unterschiede in der Ziel- und Umsetzung.[2]

## Analyse: Propagation, Exploits und Payload

Bei der Analyse von Schadsoftware konzentrierte man sich bisher auf einzelne, funktionale Komponenten im Code. Herrs Vorgehen hingegen ist ein Top-Down-Ansatz: In seinem Framework *PrEP* untersucht er die Verbreitungsmethode (*Pr*opagation method), die *E*xploits und den *P*ayload[3]. Eine ähnliche Aufteilung wird in diesem Beitrag bei der Analyse einiger Exemplare im konkreten Teil vorgenommen.

Bei der *Verbreitungsmethode* werden die Wege aufgezeigt, mit denen die Schadsoftware auf das Zielgerät gelangt. Mögliche Beispiele hierfür sind schadhafte E-Mail-Anhänge, eine kompro-

mittierte Webseite, ein infizierter USB-Stick oder auch die Auslieferung mit Hilfe eines *Droppers* (Schadcode kann gleich zusammen mit einem anderen Programm, dem *Dropper*, auf dem Zielsystem installiert oder darüber zu einem späteren Zeitpunkt nachgeladen werden).

*Exploits* sind die Codeteile, mit denen Sicherheitslücken ausgenutzt werden und die Schadsoftware sich im Zielgerät einnisten kann.

Der *Payload* bezeichnet das Kernstück der Malware, also das, was ausgeführt werden soll, sobald der entsprechende Rechner unter Kontrolle gebracht wurde. Alles ist denkbar: Datendiebstahl, Erstellen einer *Backdoor*, Überwachung einer angeschlossenen Kamera – bis hin zur Manipulation von Programmen, um Hardware zu zerstören.

Eine Trennung zwischen Exploit und Payload vorzunehmen, ist äußerst sinnvoll, wenn man sich den Entwicklungsprozess vor Augen führt. Exploits können sowohl selbst gefunden als auch von externen Quellen (z. B. im Internet) gekauft werden. Exploits öffnen die Tür für das Ausführen des Payloads – das bloße Erlangen einer Root-Shell an sich hat noch keinen Effekt. Das, was durch den Payload ausgeführt wird, ist der Grund, weswegen man in den Rechner eindringt. Ohne Exploit könnte also der Payload nicht auf den Zielrechner gelangen und seine Schadfunktion ausführen; ohne Payload hingegen wäre ein Exploit noch ohne schadhafte Folgen.

### Ausblick und Beispiele für staatliche Schadprogramme

Nun werden einige Beispiele für Milware näher beleuchtet, zunächst *Stuxnet*, der prominenteste Vertreter. Stuxnet erreichte als erste Milware weltweite Aufmerksamkeit, als sie physischen Schaden in Industrieanlagen verursachte, welche scheinbar vom Internet getrennt betrieben wurden.[4] Im Anschluss werden Stuxnets Nachfolger *Duqu* und dessen Nachfolger *Duqu 2.0* betrachtet, ebenso wie *Flame* und *Gauss*, die beide in Verbindung zu Stuxnet stehen und zusammen mit Duqu als *Cousins* von Stuxnet bezeichnet werden[4]. Da bisher nur Milware aus dem angelsächsischen Raum bzw. *dem Westen* behandelt wurde, folgt die Betrachtung einer Milware aus dem osteuropäischen Raum (*Red October)*, ehe der Überblick mit dem erst 2014 entdeckten *Regin* abgerundet wird. Für alle sieben liegen entweder akademische Untersuchungen vor oder sie wurden von Teams analysiert und dokumentiert, die für Hersteller von Anti-Virus-Produkten arbeiten; die Information aller nachfolgenden Besprechungen basiert auf diesen Untersuchungsergebnissen. Sofern Information nicht gesichert oder spekulativer Natur ist, wird dies klar gekennzeichnet. Dateien von Stuxnet stehen außerdem im Internet auf der Plattform *archive.org* zum Download bereit[5] und könnten von jedermann untersucht werden; die Binärdateien anderer Milware sind nicht ohne weiteres im Internet zu finden.

## 1 Stuxnet

Im Juni 2010 wurde von Mitarbeitern der belarussischen Firma *VirusBlokAda* eine Schadsoftware entdeckt, welche sie als *RootkitTmphider* bezeichneten; wenig später wurde immer mehr über den Computerwurm bekannt, der fortan *Stuxnet* genannt wurde. Am 30. September 2010 veröffentlichte *Symantec* ein – inzwischen mehrfach überarbeitetes – Dossier über Stuxnet, in dem die bisherige Faktenlage aus technischer Sicht zusammengefasst wurde.[6] Dieses Dossier, basierend auf Analysen des Wurms, ist auch im akademischen Bereich die Hauptquelle für die technische Funktionsweise von Stuxnet und dient als Referenz für die technischen Hintergründe, wie sie in diesem Abschnitt skizziert werden. Aufgrund der unzähligen Infektionen im Iran und den dortigen Schäden gilt als gesichert, dass die Angriffsziele von Stuxnet die iranische Urananreicherungsanlage in Natanz und das Kernkraftwerk in Buschehr waren.

### Grundlegender Aufbau der Anlagen

Um die Funktionsweise dieses Wurms nachvollziehen zu können, wird zunächst erläutert, wie solche Anlagen grundsätzlich aufgebaut sind. In heutigen Industrieanlagen werden technische Prozesse mittels Computerprogrammen gesteuert und überwacht, sogenannte *SCADA*-Systeme (*Supervisory Control and Data Acquisition*). Klassischerweise bestehen Automatisierungsanwendungen aus folgenden Komponenten: Sensoren und Aktoren messen bzw. manipulieren den technischen Prozess, die Information wird einem Controller zugeführt. Ferner existiert ein *Human Machine Interface* (HMI), das eine Schnittstelle zu Mitarbeitern vor Ort darstellt. Des Weiteren kann eine Schnittstelle zu Fernwartungszwecken bereitstehen, so dass ein externer Zugang zu Sensoren, Aktoren oder Controller vorhanden ist.

Im konkreten Fall war Siemens' System *Simatic* (*Si*emens Auto*matic*) im Einsatz: Dessen Herzstück bildet die Speicherprogrammierbare Steuerung (SPS). Die Software zur Programmierung der SPS heißt *STEP 7* und die Geräte, die zur Programmierung verwendet werden, sind *Field PGs* (SIMATIC Field PG). Siemens verwendet als Prozessleitsystem *SIMATIC PCS 7* und stellt überdies Bedien- und Beobachtungssysteme *SIMATIC HMI* zur Verfügung. Auf diesen ist die Visualisierungssoftware *WinCC* installiert, welche unter Windows läuft.

### Vorgeschichte: Stuxnet 0.5

Anfang 2013 tauchte eine weitere Version von Stuxnet auf, welche fortan als *Stuxnet 0.5* bezeichnet wurde. Es handelte sich um die älteste gefundene Version, welche bewies, dass Stuxnet älter war, als man zunächst angenommen hatte. Symantec analysierte die Version und veröffentlichte die neuen Erkenntnisse in einem Whitepaper.[7]

Bereits im November 2005 wurde demnach ein *Command-and-Control-Server* registriert und spätestens seit dem 15. November 2007 war Stuxnet 0.5 in Umlauf. In dieser Vorversion wurden Ventil-Steuerungen sabotiert, um die Verteilung von Uranhexafluorid-Gas zu kontrollieren. Der Druck in den Zentrifugen-Kaskaden wurde um das Fünffache erhöht, um so die Gerätschaften zu zerstören.[8]

Stuxnet 0.5 wurde so programmiert, dass es ab dem 4. Juni 2009 keine weiteren Systeme infizierte, so dass ab Juni 2009 die

Variante 1.001 aktiv werden konnte, die man zunächst für die erste Stuxnet-Version hielt. Im Folgenden geht es nun um die *Hauptversionen* (1.x) von Stuxnet.

### Infektionswege und Exploits

Stuxnet hatte mehrere Möglichkeiten, sich einzunisten und zu verbreiten. Insgesamt wurden drei Schichten befallen:

1. Windows-Betriebssystem
2. Siemens PCS-7, WinCC und STEP 7
3. Siemens S7 SPS

Stuxnet wurde ursprünglich durch USB-Sticks in Umlauf gebracht. Hierfür wurde ein erster *Zero-Day-Exploit* verwendet, welcher das fehlertolerante Parsen der *autorun.inf* ausnutzte. Ferner soll im Iran ein Mitarbeiter einen infizierten Stick absichtlich in die Anlage gebracht haben.[9] Nach der Erstinfektion wurde nun – je nach System – ein zweiter Zero-Day-Exploit für eine Privilegeskalation genutzt. Bis zur Version Windows XP SP2 nutzte Stuxnet dazu einen Fehler im Kernel-Mode-Treiber *win32k.sys*, in neueren Versionen eine Lücke im Task-Scheduler. Danach sollte der Schadcode in installierte Antiviren- und Windows-Systemdienste injiziert werden, ehe die eigentliche Installation in einem eigenen, vom kompromittierten System als vertrauenswürdig eingestuften Prozess ausgeführt wurde. Damit das *Rootkit* einen Neustart überleben konnte, wurden Treiber-Dateien mittels gestohlener Zertifikate der Firmen *JMicron* und *Realtek* eingeschleust. Das Betriebssystem prüft die Unterschriften, um zu verhindern, dass sich Schadsoftware im Systeminneren installieren kann. Durch die beiden gestohlenen – aber eben korrekten – Signaturen hielt das Betriebssystem Stuxnet für unschädlich und ließ die Installation zu. Anschließend verbreitete sich Stuxnet im LAN und aktualisierte sich gegenseitig. Dies war auch ohne Internetverbindung möglich. Neben Peer-to-Peer-Updates verbreitete sich Stuxnet auch über Dateifreigaben und einen Zero-Day-Exploit bei Microsofts Druckerfreigaben.

Auf der zweiten Ebene wurden STEP 7-Projektdateien infiziert. Die Schwachstelle lag hierbei in fest einprogrammierten Logins in der WinCC-Datenbank-Software. Hauptaktivität war ein *Hook* der Datei *s7otbxdx.dll*, einer zentralen Bibliothek, mit der die Kopplung einer SPS mit einer Step 7-Anwendung oder einem Field-PG stattfindet. Diese wurde durch die Veränderung eines Buchstabens in *s7otbx**s**x.dll* umbenannt und durch eine eigene *s7otbxdx.dll* ergänzt. Somit konnten Schreib- und Lesezugriffe zur SPS überwacht werden.

Auf dritter Ebene fand nur bei speziellen Hardwarebausteinen eine Manipulation statt, auf die hier nicht näher eingegangen werden muss. Die Stuxnet-Variante A konnte Frequenzumformern der finnischen Firma *Vacon*, die Variante B dem Hersteller *Fararo Paya* (Teheran) zugeordnet werden.[6] Frequenzumrichter regeln unter anderem die Geschwindigkeit von Motoren. Für eine noch detailliertere Analyse wird auf Symantecs Dossier verwiesen.[6]

### Payload

Wie konnte Stuxnet nun Schaden anrichten? In unregelmäßigen Abständen zwischen 13 Tagen und drei Monaten wurde die von den bereits angesprochenen Umformern einzustellende Frequenz geändert. Dies war eine grundlegende Neuerung zu Stuxnet 0.5: Anstatt die Zentrifugen durch Manipulation der Rotationsgeschwindigkeit zu zerstören, wurde der Druck in den Zentrifugen erhöht, so dass das Ergebnis nach einem Alterungsprozess aussah und nicht nach einer Zerstörung. Um die Manipulation vor den Mitarbeitern in Natanz zu verbergen, spielte die Milware eine vor der Manipulation aufgezeichnete, 21 Sekunden lange Messwert-Sequenz in einer Schleife in das Kontrollsystem ein. Im Anschluss veränderte Stuxnet nach und nach den Druck in der Anlage.[10]

### Folgen

Die internationale Atomenergieorganisation (IAEO) stellte bei Kontrollen im Iran mehrere Produktivitätseinbrüche bei den Zentrifugen-Kaskaden von Januar bis August 2009 fest, obwohl sich die Anzahl an Zentrifugen vergrößerte. Ferner mussten insgesamt knapp 1000 Zentrifugen ausgetauscht werden.[11] Es wird vermutet, dass die Störungen das Werk von Stuxnet und der Vorversion waren. Im November 2010 gab Irans Präsident Mahmoud Ahmadinejad zu, dass das iranische Atomprogramm sabotiert wurde und es bei einer begrenzten Anzahl Zentrifugen zu Problemen kam.[12]

Auch wenn die Mehrzahl der Infektionen im Iran stattfand, breitete sich Stuxnet weltweit aus. In Deutschland waren 59 Prozent der befragten Strom-, Gas- und Wasserversorger von Stuxnet befallen, im internationalen Durchschnitt „nur“ 41 Prozent.[13] Stuxnet richtete aber keinen Schaden in Anlagen an, die nicht Ziel dieser Cyberattacke waren.

### Urheber

Mit über 100.000 Zeilen Code, vier Zero-Day-Exploits, einem Windows-Rootkit, der ersten SPS-Schadsoftware, Antivirus-Umgehungstechniken, komplexer Prozessinjektion und Hooking-

**Sebastian Nemetz**

**Sebastian Nemetz** absolvierte sein Bachelor- und Masterstudium der Informatik mit dem Schwerpunkt IT-Sicherheit an der Friedrich-Alexander-Universität Erlangen-Nürnberg und arbeitet seit diesem Jahr bei einem Münchner IT-Sicherheitsunternehmen als IT-Berater und Penetrationstester.

code, zwei gestohlenen Signaturen, Netzwerkinfektionsroutinen, Peer-to-Peer-Updates und Command-and-Control-Interface war Stuxnet der bisher komplexeste Wurm, so dass seine Kosten im sieben- bis achtstelligen Bereich lagen und das Projekt in diesem Umfang nur von Staaten zu realisieren war.[14]

Als Urheber gelten die Vereinigten Staaten von Amerika und Israel. Auch wenn beide das nicht offiziell bestätigt haben, gibt es kaum Zweifel daran. Im Jahr 2013 wurde ein Verfahren gegen General James Cartwright (2007 bis 2011 stellvertretender US-Generalstabschef) eingeleitet, da er geheime Informationen über die Stuxnet-Attacke an die *New York Times* weitergegeben haben soll.[15] Dies ist ein weiteres Indiz dafür, dass die Attacke unter Beteiligung der USA ausgeführt wurde.

Dass das Ziel die Sabotage der Atomanlagen im Iran war, ist aufgrund der Vielzahl der Infektionen im Iran und den genauen Angaben über die Hardwarebausteine in den betreffenden Anlagen, auf die Stuxnet exakt zugeschnitten war, unbestritten. Neben diversen Spekulationen ist ein Indiz für die israelische Beteiligung, dass sich in Dimona baugleiche Zentrifugen befinden, an denen der Wurm hätte getestet werden können.[16]

### Neue Erkenntnisse 2016: Nitro Zeus

Durch den Dokumentarfilm *Zero Days* kamen Anfang 2016 neue Erkenntnisse über Stuxnet ans Tageslicht. Der Film behauptet, dass Stuxnet Teil eines viele Millionen schweren Programms namens *Nitro Zeus* war, dessen Ziel es war, die komplette zivile Infrastruktur des Irans zum Erliegen bringen zu können.[17] Im Falle eines Krieges hätten so Stromversorgung, Kommunikationsnetze und weitere zentrale Infrastruktur lahmgelegt und der Iran ohne militärisches Eingreifen geschwächt werden können. US-Geheimdienstler geben zudem den Israelis die Schuld an der Aufdeckung von Stuxnet, da diese den Wurm eigenmächtig verändert und eine sich aggressiver verbreitende Variante eingesetzt hätten. Recherchen der New York Times kamen zum gleichen Ergebnis und stützten die Thesen aus Zero Days.[18]

## 2 Duqu

Nun wird die Milware *Duqu* vorgestellt, welche als ein Nachfolger von Stuxnet gilt. Im September 2011 entdeckten Forscher des *Laboratory of Cryptography and System Security* (CrySyS Lab) am Department für Telekommunikation an der Budapester University of Technology and Economics eine bis dato unbekannte Schadsoftware bei einem europäischen Unternehmen, das sie beauftragt hatte, einen Sicherheitsvorfall in deren IT-Systemen zu untersuchen.[4] Sie tauften die Milware Duqu, da auf infizierten Rechnern Dateien angelegt wurden, deren Dateinamen mit *~DQ* begannen. In Folge analysierte das CrySyS Lab das gefundene Sample und teilte es mit großen Anti-Virus-Herstellern. Mitte Oktober 2011 erschien ein erster Report von *Symantec*. Die Erkenntnisse über die Funktionsweise von Duqu, auf die sich auch dieser Beitrag stützt, stammen aus den Veröffentlichungen des CrySyS Lab[19] und Symantecs Bericht[20].

### Allgemeines

Das Ergebnis einer ersten Untersuchung war die Erkenntnis, dass Duqu äußerst ähnlich zu Stuxnet aufgebaut ist, wenn man Designphilosophie, interne Strukturen und Mechanismen sowie Implementierungsdetails miteinander vergleicht. Trotz der frappierenden Ähnlichkeit zeigte sich, dass Duqu ein anderes Ziel verfolgte. Während bei Stuxnet das Erzielen physischen Schadens im Vordergrund stand, handelt es sich bei Duqu um eine informationssammelnde Milware, die zur Cyberspionage verwendet werden kann. Aufgrund der Ähnlichkeit liegt natürlich der Verdacht nahe, dass Duqu von derselben Gruppe von Leuten wie Stuxnet entwickelt wurde und sie Zugang zum Stuxnet-Quellcode hatten.

### Verbreitung und Exploits

Als Dropper-Komponente konnte ein Dokument für Microsoft Word ausfindig gemacht werden, welches einen Zero-Day-Kernel-Exploit enthielt. Der Exploit nutzte dabei – nach einer gewissen Wartezeit und sofern bestimmte Randbedingungen auf dem System stimmten – einen unbekannten Bug im Windows-Kernel beim Handling eingebetteter Schriftarten aus, für den im Anschluss an die Veröffentlichung im Dezember 2011 ein Patch bereitgestellt wurde (CVE-2011-3402).

Insgesamt gab es drei Hauptgruppen an Malware-Komponenten: Erstens einen Keylogger, außerdem jeweils eine Gruppe von Objekten, die mit dem Kerneltreiber *jminet7.sys* bzw. dem Kerneltreiber *cmi4432.sys* in Verbindung standen.

Der Keylogger enthielt eine interne, verschlüsselte DLL, welche die Keylogging-Funktionalität zur Verfügung stellte und eine Hauptanwendung, welche den Keylogger injizierte und den Logging-Prozess kontrollierte. Es tauchten zwei Varianten auf, siehe hierzu die Abschnitte über Infostealer 1 und 2, in denen sie näher erläutert werden.

Bei der zweiten Gruppe von Malware-Komponenten wurde in der Registry ein Service erstellt, der den Treiber *jminet7.sys* während des Hochfahrens lädt. Der Kerneltreiber lädt dann die Konfigurationsdaten und injiziert eine DLL namens *netp191.pnf* in einen Systemprozess. Ferner werden Konfigurationsdaten in einer verschlüsselten Datei *netp192.pnf* gespeichert. Die dritte Kategorie verhält sich ähnlich, nutzt aber die Dateien *cmi4432.pnf* und *cmi4464.pnf*.

Diese Verhaltensmuster ähneln dem von Stuxnet, zudem injiziert Duqu auch Code in die *lsass.exe*. Des Weiteren werden dieselben Hooks für *ntdll.dll* verwendet wie bei Stuxnet, und der Treiber *cmi4432.sys* hat eine valide digitale Signatur des taiwanischen Herstellers *C-Media Electronics*, von dem der Treiber aber offenkundig nicht kam – auch bei Stuxnet wurde Code mit kompromittierten taiwanischen Zertifikaten unterschrieben. Während Duqus Initialisierung werden drei Entschlüsselungsoperationen ausgeführt, auch das gleicht Stuxnet. Wie bei Stuxnet sucht auch Duqu zuerst nach bekannten Anti-Virus-Produkten, um dann eine bösartige DLL zu injizieren.

### Payload

Während also viele Komponenten von Duqu und Stuxnet gleich aufgebaut waren, war der Payload ein komplett anderer. Es ging nicht um die Zerstörung von Industrieanlagen, sondern die Gewinnung von Information über Systeme, möglicherweise für zukünftige Angriffe.

Duqu verwendet HTTP und HTTPS, um mit Command-and-Control-Servern zu kommunizieren. C&C-Server wurden unter anderem in Indien, Belgien und Vietnam gehostet und leiteten den Traffic an andere Server weiter, um so die Identifizierung und Nachverfolgung zu erschweren.

Duqu konnte ausführbare Dateien herunterladen und eigene Daten hochladen. Dabei wurde vorgegeben, .jpg-Bilder zu übertragen, an deren Ende verschlüsselte Daten angehängt wurden. Die „Verschlüsselung" erfolgte dabei durch eine Komprimierung mit *bzip2* und anschließende XOR-Verschlüsselung. Die ersten 8.192 Bytes des Bildes entsprechen dabei einer Aufnahme des Hubble-Weltraumteleskops.

Duqus Standardeinstellung war, 30 Tage lang aktiv zu sein und sich dann selbst zu deinstallieren; die Dauer konnte aber via C&C-Server verlängert werden. Im Gegensatz zu Stuxnet verbreitete sich Duqu nicht von selbst weiter. Insgesamt konnten vier zusätzlich über den C&C-Server heruntergeladene Binaries beobachtet werden:

**Infostealer 1** ist ein *Standalone Executable*, das auf kompromittierten Geräten gefunden wurde, aber in keinem anderen Executable enthalten war (also heruntergeladen wurde). Es hatte neun Hauptroutinen:[20]

- *List of running processes, account details, and domain information*
- *Drive names and information, including those of shared drives*
- *Take a screenshot*
- *Network information (interfaces, routing tables, shares list, etc.)*
- *Keylogger*
- *Window enumeration*
- *Share enumeration*
- *File exploration on all drives, including removable drives*
- *Enumerate computers on the domain through NetServerEnum*

Das Logfile speichert Aufzeichnungen der folgenden Felder: *Type, Size, Flags, Timestamp, Data.*[20]

**Infostealer 2** ähnelt der ersten Variante, ist aber eine DLL, aktueller (August vs. Mai 2011) und hat weniger Funktionalität als Infostealer 1. Die sieben Features sind:[20]

- *List of running processes, plus account and domain*
- *List drive names and information, including shared drives*
- *Screenshot*
- *Network information (interfaces, routing tables, and shares list)*
- *Windows enumeration*
- *Share enumeration*
- *Share browse*

*Keylogger, File exploration on all drives, including removable drives* und *Domain's servers enumeration (using NetServerEnum)* wurden entfernt.[20]

**Reconnaissance module**, ein *Aufklärungsmodul*, liefert nur einige wenige Informationen über das System (Teil einer Domain?, PID, Session ID, Windows-Ordner, Temp-Ordner, Betriebssystemversion, Architektur, Kontoname, Netzwerkadapter und Zeitinformationen).

**Lifespan extender module** konnte die *Lebensdauer*, nach der sich Duqu von selbst entfernte, verlängern.

Für eine noch ausführlichere Analyse wird auf die eingangs genannten Berichte[19,20] verwiesen.

## 3 Duqu 2.0

Auch Duqu wurde weiterentwickelt, und so fand Kaspersky im Jahr 2015 eine überarbeitete Version. *Duqu 2.0* ist Teil einer Spionagekampagne gegen die Kaspersky Labs, Hotels und Konferenzeinrichtungen bei den Verhandlungen der E3+3-Staaten zur Beilegung des Streits um das iranische Nuklearprogramm, an deren Ende im Juli 2015 die Einigung mit dem Iran zum *Joint Comprehensive Plan of Action* stand.[2,21]

Da direkt betroffen, untersuchte Kaspersky die Milware ausgiebig. Kasperskys Analysen und die des CrySyS Labs kommen zu dem Ergebnis, dass die Software eng mit Duqu verwandt ist und sich lediglich der Payload geändert hat, die zugrundeliegende Architektur aber ähnlich geblieben ist.[22,23] Diese Berichte dienen als Quelle für die Aussagen über Duqu 2.0.

Duqu und Duqu 2.0 verwendeten ähnliche Entschlüsselungsroutinen für Strings, die in Zusammenhang mit Anti-Virus-Produkten stehen. Ebenso werden ähnliche Methoden, magische Nummern und Dateiformate für AES-verschlüsselte Dateien verwendet, sogar derselbe Bug tritt auf; der verwendete (Nicht-Standard-)CBC-Modus für die AES-Verschlüsselung war derselbe; das Logging-Modul war äußerst ähnlich mit denselben magischen Konstanten und sogar das C++-ähnliche Coding und die Kompilierungsart ähnelten sich.

### Verbreitung und Exploits

Duqu 2.0 wurde u. a. durch gezielte Phishing-E-Mails in Umlauf gebracht. Er verbreitete sich durch Microsoft-Windows-Installer-Pakete, um weitere Maschinen zu infizieren. Außerdem hatte er Module für einen *Pass-the-Hash*-Angriff innerhalb eines lokalen Netzwerks, sodass den Angreifern eine Vielzahl an Verbreitungsmöglichkeiten zur Verfügung stand. Es wurden ähnliche Exploits wie bei Duqu verwendet und eine Schwachstelle bei den Schriftarten (Windows True Type) ausgenutzt, um Administratorrechte zu erlangen.

### Payload

Der Payload von Duqu 2.0 hatte viele Möglichkeiten zum Initieren, Einfrieren und Umgehen von Intrusion-Detection-Systemen und Anti-Virus-Produkten sowie über 100 weitere konfigurierbare Module, welche als einzelne Pakte heruntergeladen werden konnten. Auszugsweise werden die Funktionalitäten der Module in Tabelle 1 aufgelistet.

Für eine ausführliche Darstellung aller Funktionen sei hier auf Kasperskys Analyse[23] verwiesen.

## 4 Flame

Im Mai 2012 analysierte unter anderem das CrySyS Lab eine neu entdeckte Milware, welche zunächst *sKyWIper* getauft und später *Flame* genannt wurde. Erste Versionen tauchten bereits 2007 in Europa, 2008 in den Vereinigten Arabischen Emiraten und 2010 im Iran auf. Flame ist ebenfalls eine informationsstehlende Schadsoftware. Wie auch Stuxnet und Duqu komprimiert und verschlüsselt Flame seine Dateien. Als Quelle für die Fähigkeiten dieser Milware dienen insbesondere die Analysen vom CrySyS Lab[24], sowie Untersuchungen des Kaspersky Labs, dessen Ergebnisse ebenfalls in das Werk des CrySyS Labs eingeflossen sind.

### Verbreitung und Exploits

Das CrySyS Lab konnte keine Dropper nachweisen, auch ist unklar, wie die Erstinfektion vonstattenging. Sobald Flame jedoch einen Rechner infiziert hatte, gab es viele Möglichkeiten, sich zu verbreiten: Es nutzt dieselben Drucker- (MS10-061) und LNK-Exploits (MS10-046) wie Stuxnet. Ferner kann es sich als Proxy für Windows Update ausgeben, so dass Rechner im Netzwerk, die Updates erhalten wollen, vom infizierten Rechner Malware geliefert bekommen. Damit dies funktionieren kann, ist eine gültige digitale Signatur notwendig. Für die Erstellung des Zertifikats wurde unter anderem ein MD5-Kollisionsangriff durchgeführt; eine detaillierte Erklärung dieses Angriffs findet sich im Bericht der CrySyS Labs.[4]

### Payload

Um einen Überblick über die Funktionalitäten zu geben, werden in Tabelle 2 die aus der Milware extrahierten Codenamen mitsamt ihrer Bedeutung aufgelistet, wie sie vom Kaspersky Lab dokumentiert und vom CrySyS Lab publiziert wurden[4,25].

Bezüglich der C&C-Kommunikation (siehe *Gator*) hat das CrySyS Lab Informationen über mehr als 50 Domainnamen und mehr als 15 IP-Adressen, die für die Kommunikation verwendet wurden; die C&C-Server wurden dabei sehr häufig gewechselt.

### Urheber

In der ersten groben Analyse kam das CrySyS Lab beim Vergleichen von Duqu und sKyWIper/Flame zu dem Schluss, dass es wohl nicht vom selben Team entwickelt wurde. Vielmehr vermuteten sie, dass die Angreifer verschiedene, unabhängige Entwicklungsteams für denselben Zweck beauftragt hatten und es sich somit um zwei verschiedene Implementierungen derselben Anforderungsspezifikation handeln könnte. Wenig später gelang dem Kaspersky Lab der Nachweis, dass in Flame und Stuxnet derselbe Code verwendet wurde, sodass man davon ausgehen kann, dass es sich um parallele Projekte derselben Gruppe handelte.[26] Die *Washington Post* zitiert Roel Schouwenberg, Senior Researcher des Kaspersky Labs:[27]

> *„We are now 100 percent sure that the Stuxnet and Flame groups worked together."*

Gegenüber der Washington Post bestätigten ehemalige hochrangige Geheimdienstoffizielle, dass die NSA, CIA und das israelische Militär bei der Entwicklung von Flame beteiligt waren. Aber weder die USA noch Israel gaben hierzu eine offizielle Stellungnahme ab.[27]

| Funktion des Moduls | Details |
|---|---|
| Password Stealer | Login-Daten (Passwörter) aus Google Chrome, Firefox, POP3/HTTP/IMAP, TightVNC, RealVNC, WinVNC3/4, Outlook, SAM, LSASS-Cache, Windows Live, .Net Passport |
| Remote Desktop Administration | Macht Screenshots, bewegt die Maus, sendet Input an den Desktop |
| Erkennung von Netzwerk-Sniffern | Erkennt z. B. Wireshark, tcpview, dumpcap und weitere |
| Umfangreiche Sammlung von System- und Userinformationen | Liste laufender Prozesse, Geräte, Userliste, TCP-Tabellen, SQL-Server-Information, verbundene Drucker, PuTTY Host-Keys und Sessions, u. v. m. |
| Pipe-Backdoor | Global sichtbare Windows-Pipe |
| Sammlung von System- und Netzwerkinformationen | Task-Scheduler-Logs, Firewall-Policies, Liste aller Systemdienste etc. |
| Erzeugen eines XML-Reports über das System (benutzt eigenes Schema) | Computername, Windows-Verzeichnisse, Liste logischer Geräte, Liste aller Dateien, Betriebssystem-Seriennummer, Domainname, Netzwerkadapter-Konfiguration (IP-Adressen, MAC, MTU, Adapterliste) |
| Vollzugriff auf Dateien | Lesen, Schreiben, Metadaten |

*Tabelle 1: Payload von Duqu 2.0 (Auszug)*

| Codename | Bedeutung |
|---|---|
| *Beetlejuice* | Listet Bluetooth-Geräte um die infizierte Maschine auf, kann Maschine in *Beacon* verwandeln |
| *Microbe* | Audioaufnahme von bereits existierenden Hardwarequellen, listet alle Multimediageräte auf, speichert Gerätekonfiguration, versucht passendes Aufnahmegerät auszuwählen |
| *Infectmedia* | Wählt eine Methode (Autorun_infector, Euphoria) zum Infizieren von Medien aus |
| *Autorun_infector* | Erzeugt malwareverseuchte *autorun.inf* und startet mit einem Open-Kommando; dieselbe Methode wurde von Stuxnet benutzt, bevor es den LNK-Exploit gab |
| *Euphoria* | Erzeugt ein als Verbindungspunkt fungierendes Verzeichnis mit *desktop.ini* und *target.link*; dient als Shortcut, um Flame zu starten |
| *Limbo* | Erzeugt Backdoorkonten mit dem Login *HelpAssistant* auf den Maschinen in der Netzwerkdomäne, sofern entsprechende Rechte vorhanden sind |
| *Frog* | Infiziert Maschinen mittels vordefinierter Benutzerkonten; einziger spezifizierter Account ist *HelpAssistant* aus der Limbo-Attacke |
| *Snack* | Überwachung der Netzwerk-Interfaces, empfängt und speichert NBNS-Pakete in einem Logfile |
| *Boot_dll_loader* | Konfiguration, die eine Liste aller zusätzlichen Module enthält, die geladen und gestartet werden sollen |
| *Weasel* | Erstellt eine Verzeichnisliste |
| *Boost* | Erstellt eine Liste „interessanter" Dateien anhand verschiedener Dateinamenmasken |
| *Telemetry* | Logging |
| *Gator* | Sobald eine Internetverbindung besteht: Verbindungsaufbau zu C&C-Server, um neue Module herunter- sowie gesammelte Daten hochzuladen |
| *Security* | Identifiziert Programme, die Flame behindern könnten (Anti-Virus-Programme, Firewalls, …) |

*Tabelle 2: Payload von Flame (Auszug)*

## 5 Gauss

Forscher des Kaspersky Labs entdeckten im Juni 2012 bei der Suche nach neuen, unentdeckten Komponenten von Flame eine Schadsoftware, welche viele Module enthält, die nach bekannten Mathematikern benannt worden sind. Das Modul, in dem die sensibelsten Daten gestohlen werden, ist das Gauss-Modul, sodass man die komplette Schadsoftware als *Gauss* bezeichnet hat. Die Erkenntnisse über Gauss stammen hauptsächlich aus den Untersuchungen des Kaspersky Labs[28], unabhängige akademische Untersuchungen gibt es nicht.

### Verbreitung und Exploits

Über die Ausbreitung ist wenig bekannt; Wurm-Eigenschaften, also das eigenständige Verbreiten, konnten nicht beobachtet werden. Das Späh-Modul auf USB-Sticks nutzt – wie Stuxnet – einen .LNK-Exploit (CVE-2010-2568). Gauss infizierte Rechner im Nahen Osten; im Gegensatz zu Flame, das sich hauptsächlich im Iran verbreitet hat, fand man die meisten (1.660 von ca. 2.500) Gauss-Infektionen allerdings im Libanon.

### Payload

Gauss nutzt eine ähnliche Codebasis wie Flame und kommuniziert über C&C-Server. Wie auch Flame ist Gauss dazu entwickelt, möglichst viel Information vom infizierten Rechner zu stehlen. Gauss entwendet dabei Zugangsdaten für verschiedene Banksysteme, soziale Netzwerke, Email-Clients und von Instant-Messaging-Accounts, indem eigene Module in verschiedene Browser injiziert werden. Dadurch werden Session Data, Cookies, Passwörter und der Browserverlauf abgefangen. Gauss hat zudem spezielle Kommandos, um Daten von libanesischen Banken (z. B. Bank of Beirut und Byblos Bank) abzufangen.[4]

Die identifizierten Module von Gauss sind in Tabelle 3 mit Codenamen und Bedeutung aufgelistet. Für ein Modul gibt es mehrere Bezeichnungen (sowohl *Kurt* als auch *Godel*).

| Modulname | Bedeutung |
|---|---|
| *Cosmos* | Sammelt Informationen über CMOS, BIOS |
| *Kurt, Godel* | Infiziert USB-Laufwerke mit Modul, das Daten stiehlt |
| *Tailor* | Sammelt Informationen über Netzwerk-Interfaces |
| *McDomain* | Sammelt Informationen über Benutzer-Domain |
| *UsbDir* | Sammelt Informationen über die Laufwerke des Computers |
| *Lagrange* | Installiert eine eigene Schriftart (Palida Narrow) |
| *Gauss* | Installiert Browser-Plugins, welche Passwörter, Cookies und weitere Daten sammeln |
| *ShellHW* | Loader und Kommunikationsmodul |

*Tabelle 3: Payload von Gauss*

Die Konfiguration einer bestimmten Modulkombination für jedes System ist in einem Registry Key festgehalten. Diese Technik, wie auch die gesamte Konfigurationsstruktur, ist ähnlich zu der aus Stuxnet, Duqu und Flame.

Das Godel-Modul ist besonders interessant, da es mit der Stromchiffre RC4 verschlüsselt ist (im Vergleich zu Stuxnets, Duqus und Flames Verschlüsselung also äußerst stark) und der Schlüssel für die Entschlüsselung nicht in der Milware selbst enthalten ist. Stattdessen wird versucht, das Modul dynamisch zu entschlüsseln und den Schlüssel anhand von Strings in der Path-Variable und einigen Dateinamen zu errechnen. Das lässt darauf schließen, dass das Godel-Modul nur für ganz spezielle Zielrechner entworfen wurde. Auf Rechnern, bei denen der Schlüssel nicht wiederhergestellt werden kann, bleibt das Modul inaktiv.

## 6 Red October

Die Milware *Red October* ist im Gegensatz zu den vorherigen kein Verwandter von Stuxnet, sondern aus der Feder einer vollkommen anderen Gruppe. Die Spähsoftware, im Oktober 2012 von Kaspersky entdeckt, wurde von russischsprachigen Autoren entwickelt. Red October infizierte Forschungseinrichtungen und diplomatische Organisationen in Zentralasien und Osteuropa.[2] Der Name stammt von Kaspersky und ist eine Anspielung auf den Roman *The Hunt For Red October*.[29] Die Milware war mindestens seit dem Jahr 2007 aktiv.[30] Da keine Analyse von akademischen Institutionen vorliegt, entstammen alle Erkenntnisse ausschließlich Kasperskys Veröffentlichungen.[31]

### Verbreitung und Exploits

Für die Verbreitung wurde ein dreistufiges System verwendet. Auf erster Ebene dienten E-Mails mit bösartigen Anhängen und URLs als Türöffner. Bei den Anhängen handelte es sich um Dokumente für Microsoft Word und Excel, welche Sicherheitslücken (CVE-2009-3129, CVE-2010-3333, CVE-2012-0158) ausnutzen, außerdem gab es einen Exploit für Java (*Rhino*-Exploit für CVE-2011-3544). Beide Varianten luden einen Dropper für die zweite Stufe nach. Über den Dropper konnten wiederum verschiedene Module nachgeladen werden. Der Wurm konnte sich außerdem über das lokale Netzwerk ausbreiten. Insgesamt wurden vier verschiedene Exploits verwendet. Dabei handelte es sich nicht um Zero-Day-Exploits, sondern um Exploits für bereits bekannte Sicherheitslücken, die jedoch auf den Rechnern nicht gepatcht worden waren.[31]

### Payload

Der Payload wurde über den Dropper nachgeladen und von einem Loader entschlüsselt. Es standen mehr als 100 verschiedene Module mit unterschiedlicher Funktionalität zur Verfügung. Für den Download standen ab 2007 zusammengerechnet mindestens 60 C&C-Server bereit. Die Module, die durch die Backdoor installiert werden konnten, lassen sich in zwei Kategorien aufteilen:

- Offline-Module existieren als Dateien auf der lokalen Festplatte, können eigene *Registry Keys* erstellen, haben Logdateien auf der Festplatte und können selbstständig mit den C&C-Servern kommunizieren.

- Online-Module existieren nur im Speicher und werden nie auf die Festplatte gesichert; keine *Registry Keys*, Logs nur im RAM, senden Resultate an die C&C-Server.

Der Hauptzweck der Spähsoftware lag darin, möglichst viel Information zu sammeln. Dies umfasst Information über das infizierte System, Browserversionen, Dateinamen, Verzeichnisbäume, E-Mail-Verläufe, Zugangsdaten, Passwort-Hashes, angeschlossene mobile Geräte und weiteres.

Ein Modul nistete sich in den Adobe Reader und Microsoft-Office-Applikationen ein. Der Hauptzweck dieses Codes war es, eine sichere Methode zu haben, den Zugriff auf das Zielsystem wiedererlangen zu können. Das Modul erwartete ein speziell gefertigtes Dokument, das dem Opfer per E-Mail zugeschickt werden konnte und – da es keinerlei Exploit-Code enthielt – problemlos alle Sicherheitschecks überstand. Das Dokument wurde vom Modul verarbeitet und startete eine bösartige Anwendung, die an das Dokument angehängt war. Dieser Trick ermöglicht einen erneuten Zugriff auf den infizierten Rechner, wenn z. B. die C&C-Server unerwarteterweise offline gegangen sind (shutdown/takeover).

Die Spähsoftware infizierte weltweit die Rechner von Regierungen, Botschaften und Forschungseinrichtungen. Über die Urheber ist nichts weiter bekannt.

## 7 Regin

Die Spionageplattform *Regin* wurde erst durch im November 2014 veröffentlichte Recherchen und Untersuchungen von *The Intercept*, Symantec und Kaspersky bekannt.[32,33,34] Es ist unklar, seit wann Regin eingesetzt wurde, einige Teile existieren jedoch bereits seit dem Jahr 2003. Die Hauptziele lagen in der Informationsbeschaffung und der Erleichterung anderer Angriffe. Kaspersky weist darauf hin, dass Regin rein als Malware zu bezeichnen nichtzutreffend wäre, Regin sei vielmehr eine ganze Plattform.

### Aufbau, Verbreitung und Exploits

Regin ist gekennzeichnet durch seinen mehrstufigen Aufbau. Kaspersky und Symantec unterscheiden sich leicht in der Darstellung der Stufen, beim nachladbaren Payload wird von *Plugins* gesprochen. Symantec gibt sechs Stufen an:[33]

0. Dropper. Installs Regin onto the target computer

1. Loads driver (facilitates the loading of stage 2)

2. Loads driver (kernel driver, runs stage 3)

3. Loads compression, encryption, networking, and handling for an encrypted virtual file system (EVFS)

4. Utilizes the EVFS and loads additional kernel mode drivers, including payloads

5. Main payloads and data files

Ein Großteil des Codes befindet sich in verschlüsseltem Dateispeicher, so genannten Virtual File Systems (VFS). Für eine detaillierte Erklärung der Funktionsweise wird auf die technischen Berichte von Symantec und Kaspersky verwiesen.[33,34]

Über den initialen Infektionsmechanismus ist nur wenig bekannt, ebenso wenig wurden Zero-Day-Exploits gefunden.

### Payload

Die Plugins ermöglichten umfangreiche Überwachungsmaßnahmen. Das umfasst das Sammeln verschiedenster Information auf dem infizierten Rechner (Laufwerksnamen, Ordnerstrukturen, angeschlossene USB-Geräte), das Abgreifen von sensiblen Daten (Zugangsdaten, Passwörter, Dokumente, E-Mails), das Überwachen des Netzwerkverkehrs und viele weitere Features (z. B. Mausklicks, Screenshots).

Während oftmals eben dieser Datendiebstahl im Vordergrund stand, gibt es auch Fälle, bei denen Telekommunikationsanbieter angegriffen wurden, um weitere komplexe Angriffe führen zu können. Ein Plugin kann die Aktivitäten eines GSM Base Station Controller mitloggen.

### Opfer und Urheber

Laut Kaspersky sind die meisten Opfer in den folgenden Gruppen zu finden:[34]

- Telekommunikationsanbieter
- Regierungseinrichtungen
- Multinationale politische Gremien
- Finanzinstitute
- Forschungsinstitute
- Personen, die in Bereich der fortgeschrittenen Mathematik/ Kryptographie forschen

Regin war in vielen Ländern aktiv, vor allem in Russland und Saudi-Arabien, aber auch in Belgien, Deutschland und Österreich. Bekannte Opfer von Regin sind der belgische Telefonanbieter *Belgacom*, der belgische Kryptograph *Jean-Jacques Quisquater*, die EU-Kommission in Brüssel und eine Referatsleiterin im deutschen Bundeskanzleramt. Die Bundesanwaltschaft hatte wegen letzterer ein Ermittlungsverfahren gegen unbekannt eingeleitet.

Seit Anfang 2015 geht man davon aus, dass Regin ein Werkzeug der NSA und des GCHQ ist, welches von den Geheimdiensten der *Five Eyes* genutzt wird. Weder die NSA noch das GCHQ kommentierten die Veröffentlichungen.

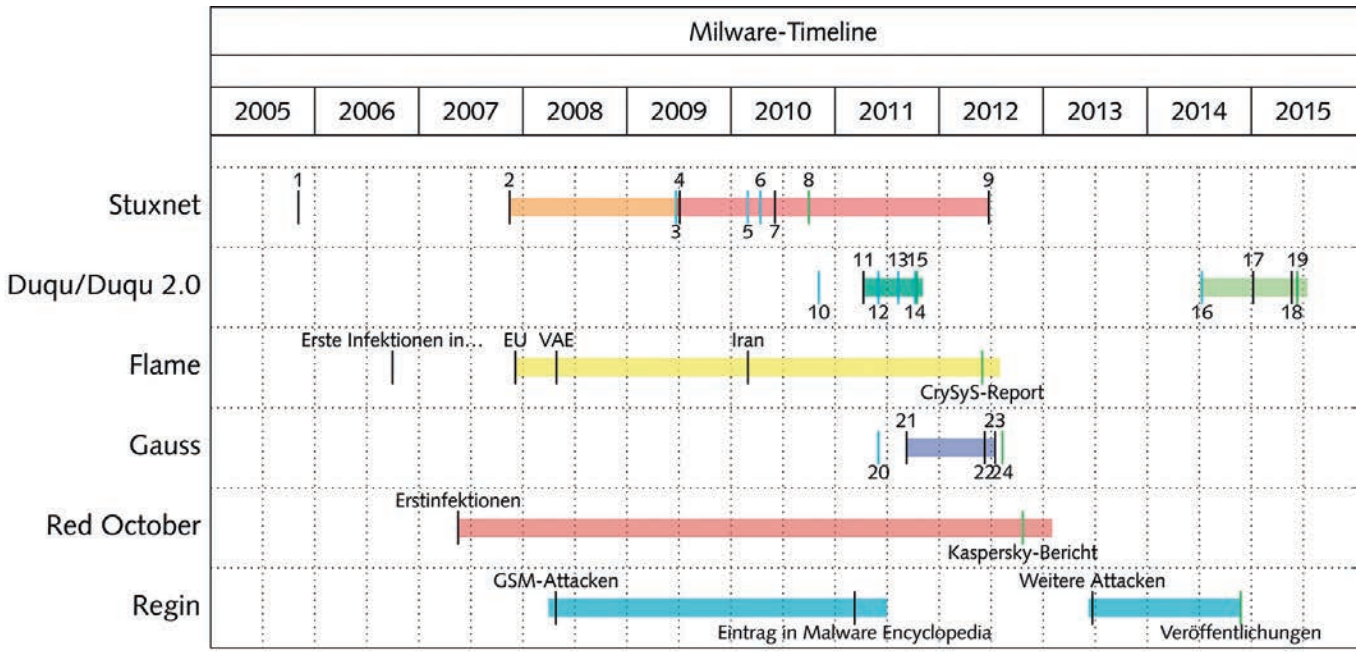
## 8 Ausblick

Abschließend werden die besprochenen sieben Beispiele für Milware in einen zeitlichen Kontext gesetzt (Abbildung 1). Die Zeitintervalle sind dabei nicht als exakte Anfangs- und Enddaten zu verstehen, sondern dienen als grobe Anhaltspunkte, in welchem Zeitraum die Milware aktiv war. Bei Stuxnet wurde der Anfang auf das Datum gelegt, an dem die Version 0.5 auf die Online-Plattform *VirusTotal* hochgeladen wurde; ab wann Stuxnet nicht mehr eingesetzt wurde, ist nicht bekannt, das Ende wurde daher für die Grafik auf den Zeitpunkt gelegt, an dem sich Stuxnet nicht mehr weiterverbreitete, auch wenn Stuxnet wohl schon früher nicht mehr benutzt wurde. Bei Duqu und Duqu 2.0 beginnt der Zeitraum ab den ersten dokumentierten Angriffen und endet kurze Zeit nach den Veröffentlichungen. Flame war seit 2007 nachgewiesenermaßen in Europa aktiv, bei der Entdeckung 2012 wurde der bisherige Einsatzzeitraum auf fünf bis acht Jahre geschätzt.[24] Gauss' früheste bekannte Infektionen waren im September 2011, im Juli 2012 gingen die C&C-Server offline. Red Octobers Einsatz ist ab Mai 2007 gesichert, Anfang 2013 war die Milware noch immer in Gebrauch.[30] Die Plattform Regin war seit mindestens 2008 im Einsatz, wurde 2011 vom Netz genommen und tauchte 2013 in einer neuen Version wieder auf. Ob Regin auch nach den Veröffentlichungen Ende 2014 noch genutzt wurde, ist nicht bekannt.

Welche Auswirkungen hat nun die Zunahme an Milware? Herr beobachtete *Trickle-down-Effekte* von Milware zu Malware.[2] Propagierungsmethoden und Exploits, die für Milware entwickelt wurden, finden Einzug bei Malware-Autoren.[35] Auch wird Code (z. B. von Duqu und Red October) von kriminellen Vereinigungen in ihrer Malware wiederverwendet. Indem Staaten durch eigene oder ausgelagerte Großprojekte ihre Angriffsstärke erhöhen, finanzieren sie letztlich indirekt auch Forschungs- und Entwicklungseinrichtungen für nicht staatliche Gruppen, was die Schere zwischen den Fähigkeiten der Angreifer und der Verteidiger noch weiter öffnet.

Auf dem Exploit-Markt können die staatlichen finanziellen Mittel die Verteidiger ausstechen. Ferner führt die staatliche Präsenz dazu, dass es mehr und mehr Anbieter für Schwachstellen in weit verbreiteter, kommerzieller Software gibt. Milware kann zur treibenden Kraft werden, was die Komplexität und Vielfalt schadhafter Software und deren Komponenten betrifft. Staaten konkurrieren bereits jetzt um die effektivsten Spionagetools und Cyberwaffen. Bei all diesen Betrachtungen muss man sich stets vor Augen führen, dass Staaten – im Gegensatz zu sonstigen Malware-Autoren – weitgehend immun gegen strafrechtliche Verfolgung sind und somit bei Entwicklung und Einsatz von Milware weitreichende Freiheiten haben, ohne Konsequenzen fürchten zu müssen.

Abschließend lässt sich festhalten, dass Milware eine neue Kategorie von bösartiger Software darstellt, bei der sich aufgrund der staatlichen Entwicklung die Prioritäten (Verwendung bei nationalen Strategien, als taktisches Mittel auf dem „Schlachtfeld" oder zur Spionage) und Komplexität fundamental von denen nicht staatlicher Gruppen unterscheiden. Diese Unterschiede bedingen, dass konventionelle Annahmen aus der Informationssicherheit auf den Prüfstand gehören. Schafft es die IT-Sicherheits-Community nicht, eine Unterscheidung zwischen Milware




| | | | | | |
|---|---|---|---|---|---|
| 1 | Registrierung der C&C-Server | 9 | Infection stop date: v1.x | 17 | Weitere Attacken |
| 2 | v0.5 bei VirusTotal hochgeladen | 10 | Treiber kompiliert, erste Varianten | 18 | Analysebeginn |
| 3 | Main binary compile timestamp: v1.001 | 11 | Erste Attacken | 19 | Kaspersky-Report veröffentlicht |
| 4 | Infection stop date: v0.5 | 12 | Infostealer 1 kompiliert | 20 | Einige Dateien kompiliert |
| 5 | Main binary compile timestamp: v1.100 | 13 | Infostealer 2 kompiliert | 21 | Früheste bekannte Infektionen |
| 6 | Main binary compile timestamp: v1.101 | 14 | Weitere Module kompiliert | 22 | Nachforschungen beginnen |
| 7 | Entdeckung von Stuxnet | 15 | CrySyS Report | 23 | C&C-Server gehen offline |
| 8 | Symantec-Dossier | 16 | Einige Module kompiliert | 24 | Kaspersky-Bericht |

*Abbildung 1: Timeline mit den aktiven Zeiträumen und Meilensteinen verschiedener Milware*

und Malware zu vollziehen und entsprechende Konsequenzen zu ziehen, wird es schwierig, auf diese neue, andersartige Bedrohung angemessen reagieren zu können.

---

*Nachwort der Redaktion: Kurz vor Übernahme dieses Textes in den Layoutprozess erreichte uns ein Hinweis des Autors auf einen tagesaktuellen Vorgang, der den oben angesprochenen Trickle-down-Effekt plastisch vor Augen führt:*

> *Hackers exploiting malicious software stolen from the National Security Agency executed damaging cyberattacks on Friday that hit dozens of countries worldwide […] They then quickly spread through victims' systems using a hacking method that the N.S.A. is believed to have developed as part of its arsenal of cyberweapons. […] The attacks on Friday appeared to be the first time a cyberweapon developed by the N.S.A., funded by American taxpayers and stolen by an adversary had been unleashed by cybercriminals against patients, hospitals, businesses, governments and ordinary citizens. […]*[36]

## Referenzen


1 *Herr T (2015) The Rise of Milware. Cyber Security Policy & Research Institute, The George Washington University, 2.3.2015, http://www2.seas.gwu.edu/~cspri/blog/2015/3/2/the-rise-of-milware.html*

2 *Herr T, Armbrust E (2015) Milware: Identification and Implications of State Authored Malicious Software. New Security Paradigms Workshop, Twente, Niederlande, 8.-11.9.2015, S. 29–43. https://ssrn.com/abstract=2569845*

3 *Herr T (2014) PrEP: A Framework for Malware & Cyber Weapons. The Journal of Information Warfare 13(1):87–106. https://ssrn.com/abstract=2343798*

4 *Bencsáth B, Pék G, Buttyán L, Félegyházi M (2012) The Cousins of Stuxnet: Duqu, Flame, and Gauss. Future Internet 4(4):971–1003, http://www.mdpi.com/1999-5903/4/4/971*

5 *Best M (2015) Stuxnet code. archive.org, 12.9.2015, https://archive.org/details/Stuxnet*

6 *Falliere N, O'Murchu L, Chien E (2011) W32.Stuxnet Dossier. Symantec Security Response, Version 1.4, Feb. 2011, https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf*

7 *McDonald G, O'Murchu L, Doherty S, Chien E (2013) Stuxnet 0.5: The Missing Link. Symantec Security Response, Version 1.0, 26.2.2013,*

*http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/stuxnet_0_5_the_missing_link.pdf*

8 *Beer K (2013) Stuxnet 0.5: Der Sabotage-Wurm ist älter als gedacht. heise Security, 27.2.2013, http://heise.de/-1812154*

9 *Eikenberg R (2012) Innenangreifer half bei Stuxnet-Infektion. heise Security, 13.4.2012, http://heise.de/-1520408*

10 *Langner R (2013) To Kill a Centrifuge: A Technical Analysis of What Stuxnet's Creators Tried to Achieve. The Langner Group, Nov. 2013, https://www.langner.com/wp-content/uploads/2017/04/To-kill-a-centrifuge.pdf*

11 *Albright D, Brannan P, Walrond C (2010) Did Stuxnet Take Out 1,000 Centrifuges at the Natanz Enrichment Plant? Institute for Science und International Security, 22.12.2010, http://isis-online.org/uploads/isis-reports/documents/stuxnet_FEP_22Dec2010.pdf*

12 *BBC News (2010) Iran says nuclear programme was hit by sabotage. BBC online, 29.11.2010, http://www.bbc.co.uk/news/world-middle-east-11868596*

13 *SPIEGEL ONLINE (2011) Deutsche Energieversorger anfällig für Computerwurm Stuxnet. DER SPIEGEL 16/2011, 16.4.2011, http://www.spiegel.de/spiegel/vorab/a-757472.html*

14 *Rieger F (2010) Der digitale Erstschlag ist erfolgt. Frankfurter Allgemeine Zeitung, 22.9.2010, www.faz.net/aktuell/feuilleton/debatten/digitales-denken/t-1578889.html*

15 *Wilkens A (2013) Stuxnet: Berichte über weiteren Geheimnisverrats-Fall in den USA. heise online, 28.6.2013, http://heise.de/-1902235*

16 *Broad WJ, Markoff J, Sanger DE (2011) Israeli Test on Worm Called Crucial in Iran Nuclear Delay. The New York Times, 15.1.2011, http://www.nytimes.com/2011/01/16/world/middleeast/16stuxnet.html*

17 *Binsch J (2016) Codename „Nitro Zeus": Vom Plan, Iran komplett lahmzulegen. Süddeutsche Zeitung, 18.2.2016, http://www.sueddeutsche.de/digital/s-1.2870281*

18 *Sanger DE, Mazzetti M (2016) U.S. Had Cyberattack Plan if Iran Nuclear Dispute Led to Conflict. The New York Times, 16.2.2016, http://www.nytimes.com/2016/02/17/world/middleeast/us-had-cyberattack-planned-if-iran-nuclear-negotiations-failed.html*

19 *Bencsáth B, Pék G., Buttyán L, Félegyházi M (2011) Duqu: A Stuxnet-Like Malware Found in the Wild. Technical Report Version 0.93, CrySyS Lab, Budapest, 14.10.2011, https://www.crysys.hu/publications/files/bencsathPBF11duqu.pdf*

20 *Symantec Security Response (2011) W32.Duqu: The precursor to the next Stuxnet. Version 1.4, 23.11.2011, https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_duqu_the_precursor_to_the_next_stuxnet.pdf*

21 *Auswärtiges Amt (2016) Konflikt um das iranische Atomprogramm. 20.1.2016, http://www.auswaertiges-amt.de/DE/Aussenpolitik/RegionaleSchwerpunkte/NaherMittlererOsten/04_Iran/Iranisches-Nuklearprogramm_node.html*

22 *Bencsáth B, Ács-Kurucz G, Molnár G, Vaspöri G, Buttyán L, Kamarás R (2015) Duqu 2.0: A comparison to Duqu. Technical Report Version 1.0, CrySyS Lab, Budapest, 10.6.2015, http://www.crysys.hu/duqu2/duqu2.pdf*

23 *Kaspersky Lab (2015) The Duqu 2.0: Technical Details. Version 2.1, 11.6.2015. https://securelist.com/files/2015/06/The_Mystery_of_Duqu_2_0_a_sophisticated_cyberespionage_actor_returns.pdf*

24 *sKyWIper Analysis Team (2012) sKyWIper (a.k.a. Flame a.k.a. Flamer): A complex malware for targeted attacks. Technical Report Version 1.05, CrySyS Lab, Budapest, 31.5.2012, https://www.crysys.hu/skywiper/skywiper.pdf*

25 *Gostev A (2012) Flame: Bunny, Frog, Munch and Beetlejuice… Kaspersky Lab, 30.5.2012, https://securelist.com/blog/incidents/32855*

26 *Kaspersky Lab (2012) Resource 207: Kaspersky Lab Research proves that Stuxnet and Flame developers are connected. Press Release, 11.6.2012, http://newsroom.kaspersky.eu/fileadmin/user_upload/en/Images/Lifestyle/20120611_Kaspersky_Lab_Press_Release_Flame_Stuxnet_cooperation_final_-_UK.pdf*

27 *Nakashima E, Miller G, Tate J (2012) U.S., Israel developed Flame computer virus to slow Iranian nuclear efforts, officials say. The Washington Post, 19.6.2012, www.washingtonpost.com/world/national-security/u/2012/06/19/gJQA6xBPoV_story.html*

28 *Global Research & Analysis Team (GReAT) (2012) Gauss: Abnormal Distribution. Kaspersky Lab, 9.8.2012, https://securelist.com/analysis/publications/36620*

29 *Clancy T (1984) The Hunt for Red October. Naval Institute Press, Annapolis, MD*

30 *Global Research & Analysis Team (GReAT) (2013) "Red October" Diplomatic Cyber Attacks Investigation. Kaspersky Lab, 14.1.2013, https://securelist.com/analysis/publications/36740*

31 *Global Research & Analysis Team (GReAT) (2013) "Red October". Detailed Malware Description 1. First Stage of Attack. Kaspersky Lab, 17.1.2013, https://securelist.com/analysis/publications/36830*

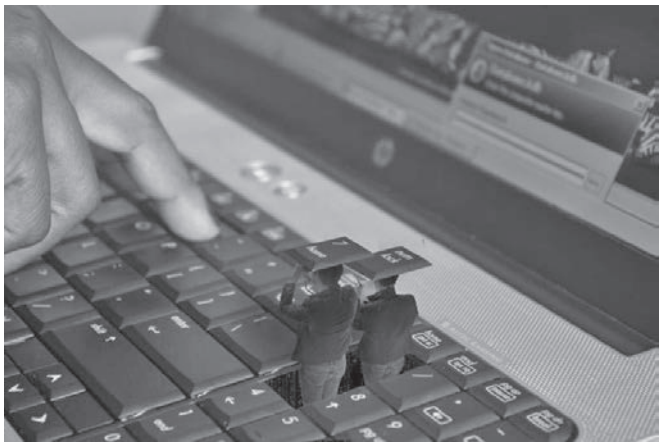
32 *Marquis-Boire M, Guarnieri C, Gallagher R (2014) Secret malware in European Union attack linked to U.S. and British intelligence. The Intercept, 24.11.2014, https://theintercept.com/2014/11/24/secret-regin-malware-belgacom-nsa-gchq/*

33 *Symantec Security Response (2015) Regin: Top-tier espionage tool enables stealthy surveillance. Version 1.1, 27.8.2015, http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/regin-analysis.pdf*

34 *Kaspersky Lab (2014) The Regin platform: Nation-state ownage of GSM networks. Version 1.0, 24.11.2014, https://securelist.com/files/2014/11/Kaspersky_Lab_whitepaper_Regin_platform_eng.pdf*

35 *Shamir U (2014) The Case of Gyges, the Invisible Malware: Government-Grade now in the Hands of Cybercriminals. Sentinel Labs Intelligence Report, Juli 2014, https://archive.org/details/pdfy-58MYDOIbvKzM8O1H*

36 *Perlroth N, Sanger DE (2017) Hackers Hit Dozens of Countries Exploiting Stolen N.S.A. Tool. The New York Times, 12.5.2017, https://www.nytimes.com/2017/05/12/world/europe/uk-national-health-service-cyberattack.html*



*keyloggers, Foto: Robbert van der Steeg, CC BY-SA 2.0*