Analyse und Vergleich von BckR2D2-I und II

Andreas Dewald[1], Felix C. Freiling*[2], Thomas Schreck[2], Michael Spreitzenbarth[2], Johannes Stüttgen[2], Stefan Vömel[2], und Carsten Willems[3]

[1]Universität Mannheim
[2]Friedrich-Alexander Universität Erlangen-Nürnberg
[3]Ruhr-Universität Bochum

Abstract: Im Oktober 2011 erregte die Veröffentlichung von Details über die inzwischen meist als *BckR2D2* bezeichnete Schadsoftware öffentliches Aufsehen. Mitglieder des *Chaos Computer Club e.V.* veröffentlichten einen ersten Bericht über die Funktionsweise des Trojaners, dem weitere Analysen folgten. In dieser Arbeit geben wir einen Überblick über die bislang veröffentlichen Einzelberichte und über die verschiedenen Komponenten der Schadsoftware sowie deren Funktionsweise. Hierzu präsentiert diese Arbeit die wesentlichen Ergebnisse einer ausführlichen Analyse aller Komponenten des Trojaners und geht insbesondere auf Unterschiede zwischen den beiden bislang bekannten Varianten BckR2D2-I und II ein. Ziel dieser Arbeit ist auch die kritische Überprüfung der von anderen Autoren getroffenen Aussagen über die Schadsoftware.

1 Einleitung

Im Laufe des Jahres 2011 wurden verschiedene Versionen der inzwischen als *BckR2D2* bezeichneten Schadsoftware bekannt. Eine erste Analyse wurde Anfang Oktober 2011 durch den *Chaos Computer Club e.V.* (CCC) in Form einer Presseerklärung sowie eines dazugehörigen 20-seitigen Berichts veröffentlicht [Cha11d, Cha11b]. In diesem werden einzelne Programmfunktionen und insbesondere der eingesetzte Verschlüsselungsmechanismus detailliert dargestellt. *F-Secure*, ein bekannter Hersteller von Anti-Viren-Software, berichtete wenige Tage später in seinem Blog über einen so genannten *Dropper* [FS11], welcher den eigentlichen Trojaner installiert. Dieser wurde offenbar bereits im Dezember 2010 beim Onlinedienst *VirusTotal*[1] hochgeladen und beinhaltet eine Installationsroutine zur Einrichtung der verschiedenen Programmkomponenten der Schadsoftware auf einem System. Knapp zwei Wochen später veröffentlichte der CCC einen Bericht über die Unterschiede zwischen beiden Varianten [Cha11e, Cha11c]. Der Fokus dieser Analyse liegt wiederum auf der Art und Weise des Einsatzes von Verschlüsselungstechniken. Eine weitere, kurze Analyse des Droppers stammt von Tillmann Werner [Wer11]. Diese Analysen, vor allem die des CCC, erregten ein umfangreiches öffentliches Aufsehen. Eine kritische Würdigung dieser Analysen sowie eine detaillierte Aufstellung der Quellengenese existiert jedoch bisher noch nicht.

*Kontaktautor, Kontaktadresse: Am Wolfsmantel 46, 91058 Erlangen, Deutschland.

[1] http://www.virustotal.com/

Ausgangspunkt unserer Arbeit war die von uns als *BckR2D2-I* bezeichnete Variante der Schadsoftware. Diese wurde uns von Rechtsanwalt Patrick Schladt zur Verfügung gestellt und stammt von der Festplatte eines seiner Mandanten. Diese Version wurde bereits 2009 eingesetzt und stellt das bislang älteste den Autoren bekannte Exemplar der Trojaner-Familie dar. Im Oktober 2011 veröffentlichten Mitglieder des CCC ebenfalls eine Variante dieses Trojaners [Cha11d], die auf den ersten Blick nicht mit der Version *BckR2D2-I* übereinstimmte. Nach eingehender Analyse gelangten wir jedoch zu der Einsicht, dass es sich bis auf minimale Modifikationen um die gleiche Variante der Schadsoftware handelt. Daher bezeichnen wir diese als *BckR2D2-I'*.[2] Zusätzlich wurde uns eine Version des Droppers [FS11, Cha11e, Wer11] von einem Hersteller von Antiviren-Software zur Verfügung gestellt. Wir bezeichnen diese Version als *BckR2D2-II*. Abbildung 1 veranschaulicht die Quellengenese, also die Herkunft der unterschiedlichen Versionen der Schadsoftware sowie die der jeweiligen Version zugehörigen Komponenten.

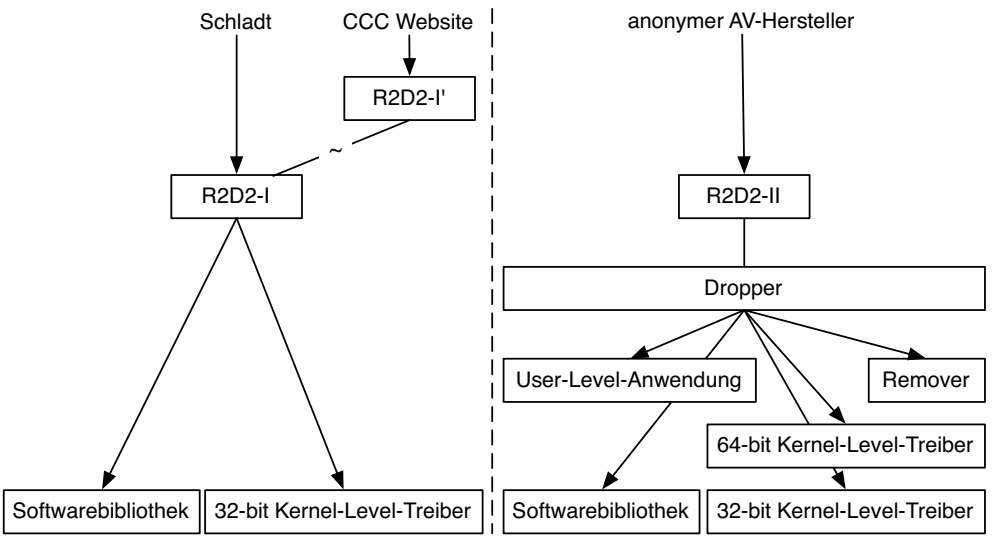

Abbildung 1: Herkunft und Komponenten der unterschiedlichen Versionen von *BckR2D2*.

Ziel dieser Arbeit ist es, einen unabhängigen Überblick über die beiden bislang bekannten, unterschiedlichen Varianten der Schadsoftware zu liefern und einen Vergleich untereinander zu ziehen. Ein besonderes Augenmerk gilt hierbei potenziellen Erweiterungen oder Einschränkungen des Funktionsumfanges sowie Verbesserungen zwischen den verschiedenen Versionen. Hierzu unterzogen wir beide Varianten der Schadsoftware einer intensiven statischen Analyse. In dieser Arbeit werden die wesentlichen Ergebnisse dieser Analyse erläutert. Im Zuge dessen werden insbesondere die in unterschiedlichen Quellen [Cha11b, Cha11e, FS11, Wer11] veröffentlichten Aussagen über einzelne Varianten der Schadsoftware überprüft.

Aufgrund der Seitenlimitierung beschränken sich die Autoren in dieser Arbeit auf die Erläuterung der wichtigsten Ergebnisse. Weitere Details werden die Autoren in einem technischen

[2] So beschreibt der CCC eine Reihe zum Zwecke des Quellenschutzes vorgenommener Änderungen, die zu BckR2D2-I' führten, auf seiner Webseite [Cha11a].

Bericht [DFS+11] zur Verfügung stellen.

1.1 Gliederung der Arbeit

Die restlichen Abschnitte dieser Arbeit sind wie folgt gegliedert: Abschnitt 2 gibt einen kurzen Überblick über die einzelnen Komponenten der Schadsoftware. Eine detaillierte Analyse der Komponenten folgt in Abschnitt 3. Wichtige Unterschiede zwischen den beiden Schadsoftwareversionen sind in Abschnitt 4 näher erläutert. Eine abschließende kurze Zusammenfassung der Untersuchungsergebnisse ist Gegenstand von Abschnitt 5.

2 Überblick über die Komponenten von BckR2D2

Bislang sind insgesamt fünf verschiedene Komponenten unterschiedlicher Funktionalität bekannt, die der Schadsoftware zuzuordnen sind. Diese Komponenten liegen jedoch ausschließlich für die Variante *BckR2D2-II* vollständig vor und sind in dem bereits genannten *Dropper* [FS11] enthalten. Für die ältere Variante *BckR2D2-I* liegt lediglich der *32-bit Kernel-Level-Treiber* und die *Softwarebibliothek* vor. Ein Überblick über die betreffenden Ressourcen ist mit einer kurzen Beschreibung in Tabelle 1 dargestellt.

Bezeichnung	**Kurzbeschreibung**	**erläutert in**
Dropper	Installationsroutine zur Einrichtung der anderen Komponenten	Abschnitt 3.1
Softwarebibliothek	Bibliothek zur Überwachung verschiedener Applikationen	Abschnitt 3.2
Kernel-Level-Treiber (32-bit)	Treiber zur Bereitstellung privilegierter Operationen (benötigt Administratorrechte)	Abschnitt 3.3
Kernel-Level-Treiber (64-bit)	Treiber zur Bereitstellung privilegierter Operationen (benötigt Administratorrechte)	Abschnitt 3.3
User-Level-Anwendung	Programm als bestmöglicher Ersatz für den Kernel-Level-Treiber bei eingeschränkten Benutzerrechten	Abschnitt 3.4
Remover	Hilfsprogramm zum Löschen beliebiger Dateien	Abschnitt 3.5

Tabelle 1: Komponenten der untersuchten Schadsoftware

Neben dem bereits genannten *Dropper*, der *Softwarebibliothek* und dem *32-bit Kernel-Level-Treiber* existiert weiterhin ein *64-bit Kernel-Level-Treiber* sowie eine *User-Level-Anwendung*, welche die Funktionalitäten des *Kernel-Level-Treibers* soweit wie möglich ersetzt, falls bei der Installation der Schadsoftware keine Administratorprivilegien zur Verfügung stehen. Ein so genannter *Remover* dient zum Löschen von Dateien und zur Verwischung angefallener Spuren. Tabelle 3 im Anhang enthält eine Übersicht über die Prüfsummen und die Dateigrößen

der einzelnen analysierten Komponenten.

3 Analyse der Komponenten

In diesem Abschnitt werden die Ergebnisse der Analyse aller Komponenten der Schadsoftware in Bezug auf ihre Funktion und Implementierung erläutert.

3.1 Dropper

Die verschiedenen Komponenten der Schadsoftware in Variante *BckR2D2-II* werden mit Hilfe einer Installationsroutine auf dem Zielsystem eingebracht. Dabei handelt es sich um eine ausführbare `.exe`-Datei. Dieses Installationsprogramm enthält sämtliche Komponenten der Schadsoftware als eingebettete Ressource, die mittels einer dreistelligen Ziffernfolge im Bereich von `101` bis `118` eindeutig referenziert werden können.

Nach Aufruf der Installationsroutine versucht diese zunächst im Windows-Systemverzeichnis des Rechners eine temporäre Datei mit dem Namen `~pgp~.tmp` anzulegen, um die Schreibberechtigung für dieses Verzeichnis zu prüfen. Sofern diese Operation erfolgreich durchgeführt werden konnte, wird die temporäre Datei anschließend sofort wieder gelöscht und eine dynamische Bibliothek mit der internen Ressourcennummer `101` unter dem Namen `mfc42ul.dll` im besagten Verzeichnis abgelegt. Es handelt sich hierbei um die *Softwarebibliothek* der Schadsoftware, welche in Abschnitt 3.2 erläutert wird.

Wahrscheinlich um die Bibliothek besser vor einer Entdeckung zu schützen oder den Installationszeitpunkt zu verschleiern, werden die Erstell-, Änderungs- und Zugriffs-Zeitstempel der Datei von der namensähnlichen Bibliothek `mfc42.dll` übernommen. Letztere ist legitimer Bestandteil der *Microsoft Foundation Classes* (MFC), einer Sammlung von Klassen für die Softwareentwicklung mit C++, die üblicherweise auf Microsoft Betriebssystemen vorinstalliert ist [Mic11]. Wenn diese jedoch nicht in Version 4.2 auf dem Zielrechner vorgefunden werden kann, so erfolgt die Anpassung aller Zeitstempel der *Softwarebibliothek* auf das fest codierte Datum *04.08.2004, 12:00 Uhr*. Die bloße Betrachtung kürzlich vorgenommener Änderungen auf dem System führt also nicht zur Aufdeckung der installierten Komponente. Jedoch existiert im NTFS-Dateisystem, welches seit Windows 2000 standardmäßig eingesetzt wird, ein weiterer, meist als *ctime* bezeichneter Zeitstempel, der die letzte Änderung der zugehörigen Dateisystem-Datenstruktur angibt [Gar09]. Dieser Zeitstempel ist nicht ohne Weiteres zur Laufzeit des Systems manipulierbar. Auf Basis dieses Zeitstempels sind daher unter Umständen dennoch Rückschlüsse auf den eigentlichen Installationszeitpunkt der Schadsoftware möglich.

In Abhängigkeit der festgestellten Betriebssystemarchitektur wird im weiteren Verlauf der Installation ein 32- oder 64-bit Treiber aus der internen Ressourcentabelle geladen (Ressourcennummer `102` und `118`) und unter dem Namen `winsys32.sys` im Systemverzeichnis der Windows-Installation gespeichert. Im nächsten Schritt wird der Treiber mit Hilfe der `CreateService`-Funktion als `SERVICE_KERNEL_DRIVER` (Servicetyp: `0x00000001`) mit vollen Rechten und ohne weitere Abhängigkeiten auf dem Computer eingerichtet und

nachfolgend gestartet. Auch hier erfolgt eine Anpassung der Zeitstempel auf die bereits bei der Installation der *Softwarebibliothek* genannten Werte. Im Anschluss an diesen Vorgang initiiert das Installationsprogramm mehrere *Code Injection*-Operationen und schleust die *Softwarebibliothek* in verschiedene Prozesse ein, namentlich in den Windows-Explorer (`Explorer.exe`) sowie in zwei Prozesse der VoIP-Software *Skype* (`Skype.exe`, `SkypePM.exe`).

In einem letzten Schritt extrahiert die Routine eine weitere Ressource, die intern unter der Nummer `106` abgelegt ist und auf der Festplatte des Zielsystems unter dem Dateinamen `~acrd~tmp~.exe` in einem temporären Verzeichnis gespeichert wird. Wie die Autoren in Abschnitt 3.5 darstellen, handelt es sich dabei um ein rudimentäres Löschwerkzeug, mit dem das ursprüngliche Installationsprogramm nach Abschluss aller Maßnahmen wieder entfernt werden kann.

Sofern die oben aufgeführten Dateien nicht im Systemverzeichnis der Windows-Installation erstellt werden können, beinhaltet das Installationsprogramm einen alternativen Infektionsweg: Hierbei wird die *Softwarebibliothek* (Ressourcennummer `101`) im Applikationsverzeichnis des aktiven Benutzers abgelegt und als versteckt markiert.[3] Als Ersatz für den Systemtreiber, der aufgrund der fehlenden Rechte nicht eingesetzt werden kann, erstellt die Routine zwei versteckte und sich gegenseitig überwachende Instanzen der *User-Level-Anwendung* unter dem Namen `SkypeLauncher.exe` und `SkypePM Launcher.exe`, die über den Registrierungsschlüssel `HKCU\Software\Microsoft\CurrentVersion\ Run` standardmäßig bei jedem Start des Betriebssystems gestartet werden. Wie wir in Abschnitt 3.4 näher darstellen werden, sind diese Prozesse für die Überwachung einer Vielzahl von Anwendungsprogrammen verantwortlich.

3.2 Softwarebibliothek

Die *Softwarebibliothek* wird in mehrere laufende Prozesse injiziert. (Die hierzu genutzten Ansätze werden in Abschnitt 4.2 näher erläutert, da sie sich in den beiden vorliegenden Varianten stark unterscheiden.) In Abhängigkeit des jeweiligen Elternprozesses, in den die *Softwarebibliothek* injiziert wurde, weisen die gestarteten Threads unterschiedliche Verhaltensmuster auf. So überprüft beispielsweise der in den *Windows Explorer* eingeschleuste Code die Version und das Patchlevel des Betriebssystems.

Ein primäres Ziel der Softwarebibliothek ist offenbar die Überwachung der Software *Skype*. Hierzu registriert sich die Bibliothek über die offizielle Schnittstelle als Skype-Erweiterung, was dazu führt, dass sie über eingehende und abgehende Anrufe und Textnachrichten informiert wird. Über die *Windows Multi-Media-Library* (`winmm.dll`) wird dann im Falle eines Anrufs der Ton aufgezeichnet, mit dem freien Audiocodec `libspeex` komprimiert und an den Kontrollserver übertragen. Der *Kontrollserver* ist ein System, von welchem der Trojaner mögliche Steuerbefehle erhält bzw. an den er aufgezeichnete Daten versendet.

Um einen Informationsaustausch mit anderen observierten Programmen zu ermöglichen, werden *benannte Dateimappings* verwendet, eine Art von gemeinsamem Speicher (*Shared Memory*) unter Microsoft Windows. Die Namen dieser Mappings beginnen in *BckR2D2* stets mit der Zeichenfolge `SYS!I[PC|CP]!`, gefolgt von einer mehrstelligen Ziffernkombination, im

[3] Unter Microsoft Windows XP ist das Applikationsverzeichnis eines Benutzers unter `C:\Dokumente und Einstellungen\<Benutzername>\Anwendungsdaten` zu finden.

Fall des *Microsoft Messengers* zum Beispiel, `79029`. Die Schadsoftware ist ebenfalls in der Lage, die gesammelten Informationen über das Internet an einen Kontrollserver zu senden. Hierzu nimmt ein dedizierter Thread über den Port 443 Kontakt mit dem Server auf.

Kommando	Beschreibung	I	II
0x02	Anfertigung von Bildschirmschnappschüssen von aktiven Fenstern eines Internetbrowsers	x	
0x03	Erstellung eines dedizierten Threads, der periodische Bildschirmschnappschüsse des kompletten Bildschirms anfertigt	x	
0x04	Entfernung des Kernel-Level-Treibers, Einrichtung der Schadsoftware im Applikationsverzeichnis des Benutzers	x	x
0x05	Aktualisierung der Schadsoftware über das Netzwerk	x	x
0x06	Herunterfahren des Systems per `ExitWindowsEx()` (`EWX_SHUTDOWN`, `EWX_FORCE`)	x	x
0x07	Herunterfahren per `KeBugCheckEx()` (erzeugt einen *Blue Screen of Death*)	x	x
0x08	Abfrage der installierten Anwendungen und Programme	x	
0x09	Erstellung eines weiteren Threads, der periodische Bildschirmschnappschüsse anfertigt (Code ähnlich zu Kommando `0x03`)	x	
0x0C	Entgegennahme mehrerer (unbekannter) Parameter über das Netzwerk	x	x
0x0D	Anfertigung von Bildschirmschnappschüssen von bestimmten Fenstern im Vordergrund, unter anderem auch Internetbrowsern (ähnlich zu Kommando `0x02`)	x	
0x0E	Übertragung und Ausführung einer beliebigen Datei	x	x
0x10	Noch unbekannt	x	x
0x11	Noch unbekannt	x	x
0x12	Null-Rückgabe	x	

Tabelle 2: Unterstützte Kommandos der Schadsoftwarebibliothek in den beiden Trojaner-Varianten (x = Funktionalität vorhanden)

Der Kontrollserver kann mit Hilfe einer Reihe von Kommandobefehlen den Client zur Durchführung verschiedener Operationen anweisen. Dazu gehört insbesondere die Anfertigung von Bildschirmschnappschüssen, aber auch die Übertragung und Ausführung beliebiger Dateien. Eine Übersicht und Beschreibung der in den jeweiligen Versionen implementierten Befehle ist in Tabelle 2 aufgeführt. Wie aus dieser Tabelle ersichtlich, konnten die Kommandos `0x0C`, `0x10` und `0x11` bis zum gegenwärtigen Zeitpunkt noch nicht vollständig analysiert werden.

3.3 Kernel-Level-Treiber

Dieser Abschnitt basiert im wesentlichen auf der Analyse von *BckR2D2-I*. Die beschriebene Funktionalität ist auch in *BckR2D2-II* enthalten, jedoch enthält *BckR2D2-II* einige weitere Funktionen, deren Zweck zum gegenwärtigen Zeitpunkt unklar ist.

Der *Kernel-Level-Treiber* bietet zahlreiche Funktionen, um unter Umgehung der im User-

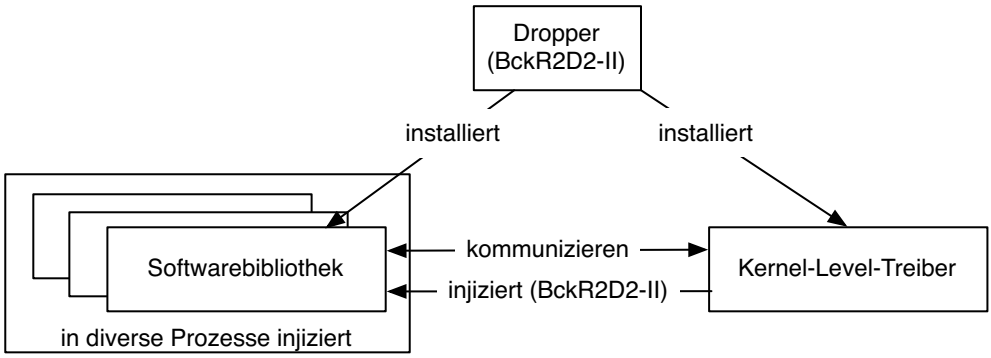

Abbildung 2: Zusammenspiel der Komponenten mit dem Kernel-Level-Treiber.

mode geltenden Sicherheitsmaßnahmen des Betriebssystems Änderungen am System vorzunehmen. Des Weiteren ist eine Keylogger-Funktionalität enthalten, die jedoch bereits in der Version *BckR2D2-I* deaktiviert ist. (Der entsprechende Code ist zwar vorhanden, wird jedoch aus dem Treiber selbst nicht angesprungen.) Außerdem beinhaltet er zwei weitere Codefragmente, die in der vorliegenden Version jedoch nicht weiter referenziert werden. Abbildung 2 zeigt eine schematische Übersicht über das Zusammenspiel des *Kernel-Level-Treibers* mit den übrigen Komponenten. Für weitere Details zum Kernel-Level-Treiber verweisen wir aus Platzgründen auf den technischen Bericht.

3.4 User-Level-Anwendung

Die Anwendung mit der Ressourcen-Nummer 107 dient als Ersatz für den Kernel-Level-Treiber. So kann die Software auch auf Systemen verwendet werden, bei denen es zum Infektionszeitpunkt nicht möglich ist, Administrator-Rechte zu erlangen. Die Anwendung erfüllt zwei Aufgaben: Zum einen überprüft sie periodisch, ob die Installation beschädigt wurde. Wird festgestellt, dass der oben genannte `Run`-Key aus der Systemregistrierung entfernt oder das Image der Anwendung von der Festplatte gelöscht wurde, generiert sie eine entsprechende Fehlermeldung und kommuniziert diese der Softwarebibliothek. Die zweite Aufgabe ist das Injizieren der Softwarebibliothek in folgende Prozesse:

- `explorer.exe`
- `SkypePM.exe`
- `yahoomessenger.exe`
- `Skype.exe`
- `msnmsgr.exe`

In Abbildung 3 ist analog zu Abbildung 2 die Einrichtung der eingesetzen Komponenten bei fehlenden Berechtigungen aufgezeigt. Dieses Szenario ist jedoch, ebenso wie der *Dropper*, nur aus *BckR2D2-II* bekannt.

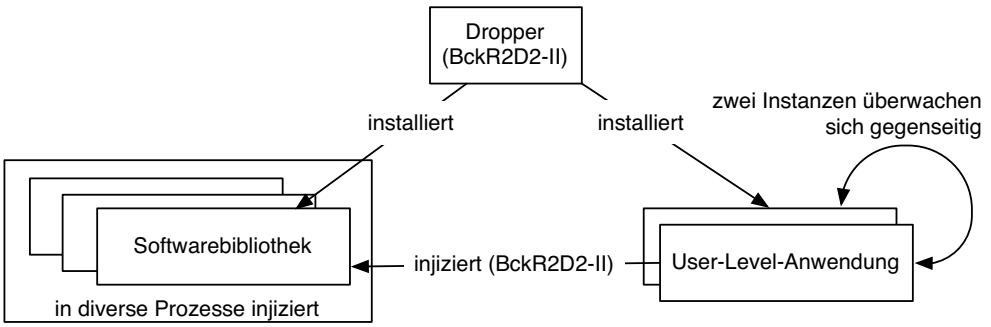

Abbildung 3: Zusammenspiel der Komponenten mit der User-Level-Anwendung.

3.5 Remover

Bei dieser Komponente der Schadsoftware, dem sogenannten *Remover*, handelt es sich um ein einfaches Löschwerkzeug, das beliebige Dateien von der Festplatte des kompromittierten Systems entfernen kann. Die zu löschende Datei wird dabei als Kommandozeilenparameter spezifiziert. Der eigentliche Löschvorgang erfolgt über den Windows-Systemaufruf `DeleteFileA()`. Diese Funktion überschreibt die eigentlichen Daten nicht, sodass es prinzipiell möglich ist, die gelöschte Datei wiederherzustellen. Nach Abschluss der Operation, die bei Nichterfolg insgesamt maximal 10 Mal wiederholt wird, vernichtet sich das Werkzeug mit Hilfe der `MoveFileExA`-Funktion beim nächsten Systemstart selbständig.

4 Vergleich von *BckR2D2-I* und *BckR2D2-II*

Im Folgenden stellen wir die beiden Varianten der Schadsoftware *BckR2D2* vergleichend gegenüber und gehen auf die wesentlichen Unterschiede ein. Neben diversen Änderungen in der *Softwarebibliothek* wurde in der Version *BckR2D2-II* auch der *Kernel-Level-Treiber* wesentlich erweitert. In dieser Version existiert erstmals neben dem *32-bit Kernel-Level-Treiber* auch eine Treiberversion für 64-bit Windows-Systeme.

4.1 Kernel-Level-Treiber

Der *Kernel-Level-Treiber* hat in der 32-bit Version von *BckR2D2-I* eine Größe von 5.376 Bytes, in Version *BckR2D2-II* bereits eine Größe von 10.112 Bytes. Anzumerken ist, dass sich die Funktionen des Treibers zwischen den Varianten *BckR2D2-I* und *BckR2D2-II* bezüglich der unterstützten Routinen und der Kommandokommunikation zur Usermode-Applikation nicht wesentlich verändert haben. In Variante *BckR2D2-II* wurde der *Kernel-Level-Treiber* jedoch dahingehend erweitert, die *Softwarebibliothek* in verschiedene Prozesse zu injizieren. Hierzu startet der Treiber einen zusätzlichen Thread, um kontinuierlich die Liste aller laufenden Prozesse zu ermitteln und den Schadcode der *Softwarebibliothek* in die folgenden Anwendungs-

und Kommunikationsprogramme zu injizieren:

- skype.exe
- paltalk.exe
- voipbuster.exe
- simplite-icq-aim.exe
- skypepm.exe
- yahoomessenger.exe
- opera.exe
- msnmsgr.exe
- x-lite.exe
- simppro.exe
- icqlite.exe
- firefox.exe
- explorer.exe
- lowratevoip.exe.

Der Kernel-Level-Treiber in *BckR2D2-I* besitzt diese Funktionalität nicht. Hier wird das Injizieren der *Softwarebibliothek* anderweitig erreicht (siehe Abschnitt 4.2). Zusätzlich zum 32-bit Treiber besitzt die Variante *BckR2D2-II* eine 64-bit Version des gleichen Kernel-Level-Treibers. Voraussetzung für das Laden dieser Treiberversion unter 64-bit Windows-Systemen ist die Signierung des Treibers mit einem digitalen Zertifikat. Der 64-bit *Kernel-Level-Treiber* aus Variante *BckR2D2-II* wurde hierzu mit einem RSA Zertifikat des fiktiven Ausstellers *Goose Cert* (Fingerprint: e5 44 5e 4a 9c 7d 24 c8 43 f0 c6 69 e2 a8 d3 a1 78 cf 7f a8) signiert. In unseren Experimenten vertraute ein standardmäßig eingerichtetes 64-bit Windows 7-System diesem Zertifikat nicht und erforderte die manuelle Bestätigung der Treiberinstallation.

4.2 Softwarebibliothek

Die *Softwarebibliothek* aus Version *BckR2D2-I* wurde den Angaben im PE-Header zufolge am 07.04.2009, 14:39:10 Uhr, kompiliert. Die Version *BckR2D2-II* hingegen wurde am 06.12.2010 um 16:23:52 Uhr erstellt und ist damit über 1 1/2 Jahre älter als seine Vorgängerversion. Obwohl eine Fälschung dieser Angaben leicht möglich ist, schätzen die Autoren eine gezielte Manipulation der Daten angesichts lediglich nur marginaler, von der Komponente angewandter Verschleierungsmaßnahmen als unwahrscheinlich ein.

In der Version *BckR2D2-I* ist aufgrund der Programmarchitektur davon auszugehen, dass die Schadsoftware durch Veränderung eines Registrierungsschlüssels in alle startenden Prozesse injiziert wird.[4] Da der Dropper für diese Version nicht vorliegt, ist dies jedoch nicht mit absoluter Sicherheit festzustellen. Die Software selbst beinhaltet aber keine entsprechende Funktionalität, sodass ein externer Eingriff für den Start zwingend notwendig ist. Wird die Bibliothek in einen Prozess geladen, führt sie initial immer einen Abgleich des Prozesses mit einer Whitelist durch. Nur wenn der Name der Image-Datei in der Liste vorhanden ist, wird die Bibliothek aktiv. In diesem Falle startet sie mehrere Threads, die Prozess-spezifische Aufgaben wahrnehmen. Die Prozessnamen in der Whitelist sind im einzelnen:

- skype.exe
- skypepm.exe
- msnmsgr.exe
- x-lite.exe
- yahoomessenger.exe
- explorer.exe

[4] Der betreffende Registrierungsschlüessel lautet HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs und veranlasst das Betriebsystem, alle hier eingetragenen Bibliotheken in den Adressraum eines jeden gestarteten Prozesses zu laden.

Die Liste der überwachten Programme wurde in der Version *BckR2D2-II* um eine weitere Anwendung, der Video-Chat-Software *Paltalk*, erweitert (`paltalk.exe`).

In Version *BckR2D2-II* wurde ein System mit der IP-Adresse `207.158.22.134` über den Port 443 als Kontrollserver kontaktiert. Diese IP-Adresse ist laut der lokalen Internetregistrierungs- und Verwaltungsorganisation *RIPE*[5] dem amerikanischen Unternehmen *Web Intellects* in Columbus, Ohio zugeordnet. In der Version *BckR2D2-II* wurde dieser Server durch einen Rechner in Deutschland ersetzt, der unter der IP-Adresse `83.236.140.90` erreichbar ist. *RIPE* hat als Inhaber der IP-Adresse die *QSC AG* registriert, welche diese wiederum an einen nicht näher genannten Kunden mit der Bezeichnung `QSC-CUSTOMER-5464944-564637` vermietet.

Bei allen untersuchten Varianten von *BckR2D2* erfolgte eine Authentifizierung der Schadsoftware gegenüber dem Kontrollserver durch Übermittlung der Zeichenkette `C3PO-r2d2-POE`, welche auch namensgebend für die Schadsoftware war. Aus diesem Grund ist es möglich, einen eigenen Kontrollserver zu präparieren und die Kontrolle über durch *BckR2D2* kompromittierte Systeme zu erhalten, wie der *CCC* bereits zeigte [Cha11b]. In Variante *BckR2D2-II* wurde das Kommunikationsprotokoll verbessert, indem sowohl beide Richtungen der Kommunikation verschlüsselt werden und eine Authentifizierung beider Kommunikationspartner erforderlich ist. Zur Verschlüsselung wird das symmetrische Verfahren AES (*Advanced Encryption Standard*) im *Electronic Codebook*-Verfahren (*ECB*) verwendet und als kryptographischer Schlüssel stets die konstante (hexadezimale) Zeichenfolge `49 03 93 08 19 94 96 94 28 93 83 04 68 28 A8 F5 0A B9 94 02 45 81 93 1F BC D7 F3 AD 93 F5 32 93` eingesetzt. Die in der Version *BckR2D2-II* neu implementierte Authentifikation des Servers gegenüber dem Client prüft lediglich, ob vor jedem Kommando vier sequentielle, fest codierte Werte (`0x95838568`, `0x0B894FAD4`, `0x8202840E`, `0x83B5F381`) übertragen werden. Der Mechanismus kann deshalb durch einen nicht-authorisierten Angreifer leicht überwunden werden. Zur Löschung des Systemtreibers und der Schadbibliothek im Systemverzeichnis des Benutzers sowie zur Einrichtung entsprechender Komponenten im Applikationsverzeichnis (Kommando `0x04`, siehe Tabelle 2) ist eine zusätzliche Authentifikationsprüfung vorgesehen. Durch Übertragung der Ziffernfolgen `0x0DEAF48F7`, `0x8474BAFD`, `0x0BAAD49F1` sowie `0x8472BCF2` kann diese ebenfalls erfolgreich überwunden werden.

Schließlich unterscheiden sich die beiden Versionen der *Softwarebibliothek* hinsichtlich ihres angebotenen Funktionsumfangs: Die Anzahl der Operationen, die vom *Kontrollserver* auf dem kompromittierten System initiiert werden können, wurde in Variante *BckR2D2-II* im Vergleich zu *BckR2D2-I* von 14 auf 8 reduziert. Die jeweiligen Kommandos wurden jedoch lediglich dadurch deaktiviert, indem sie aus der Routine zur Befehlsverarbeitung entfernt wurden. Der entsprechende Code, welcher die Funktionalität der Kommandos bereitstellt, ist jedoch weitherhin in der Bibliothek enthalten.

5 Zusammenfassung

Insgesamt konnten im Rahmen unserer Untersuchung die wesentlichen Ergebnisse der eingangs erwähnten Berichte bestätigt werden. Dies betrifft sowohl die Analyse des CCC [Cha11b] hinsichtlich der Funktionalität und des Einsatzes von Verschlüsselungsmechanismen in *BckR2D2*,

[5] http://www.ripe.net/

als auch den Vergleich der beiden Versionen der Schadsoftware [Cha11e]. Auch die durch Werner beschriebene Funktionsweise des *Droppers* [Wer11] ist nach unseren Untersuchungen zutreffend.

Literatur

[Cha11a] Chaos Computer Club. Addendum Staatstrojaner. http://www.ccc.de/de/updates/2011/addendum-staatstrojaner, 9. Oktober 2011.

[Cha11b] Chaos Computer Club. Analyse einer Regierungs-Malware. http://www.ccc.de/system/uploads/76/original/staatstrojaner-report23.pdf, 8. Oktober 2011.

[Cha11c] Chaos Computer Club. Chaos Computer Club analysiert aktuelle Version des Staatstrojaner. http://www.ccc.de/de/updates/2011/analysiert-aktueller-staatstrojaner, 26. Oktober 2011.

[Cha11d] Chaos Computer Club. Chaos Computer Club analysiert Staatstrojaner. http://www.ccc.de/de/updates/2011/staatstrojaner, 8. Oktober 2011.

[Cha11e] Chaos Computer Club. ozapftis — Teil 2: Analyse einer Regierungs-Malware. http://www.ccc.de/system/uploads/83/original/staatstrojaner-report42.pdf, 26. Oktober 2011.

[DFS+11] Andreas Dewald, Felix C. Freiling, Thomas Schreck, Michael Spreitzenbarth, Johannes Stüttgen, Stefan Vömel und Carsten Willems. Analyse und Vergleich von BckR2D2-I und II. Bericht CS-2011-08, Friedrich-Alexander Universität Erlangen-Nürnberg, Department Informatik, 2011.

[FS11] F-Secure. More Info on German State Backdoor: Case R2D2. http://www.f-secure.com/weblog/archives/00002250.html, 11. Oktober 2011.

[Gar09] Simson L. Garfinkel. Automating Disk Forensic Processing with SleuthKit, XML and Python. In *Fourth International IEEE Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE)*, Seiten 73–84. IEEE Computer Society, 2009.

[Mic11] Microsoft Corporation. MFC Reference. http://msdn.microsoft.com/de-de/library/d06h2x6e.aspx, 2011.

[Wer11] Tillmann Werner. Federal Trojan's got a "Big Brother". http://www.securelist.com/en/blog/208193167/Federal_Trojan_s_got_a_Big_Brother, 18. Oktober 2011.

A Übersicht über Prüfsummen und Dateigrößen

Tabelle 3 enthält eine Übersicht über die Prüfsummen und die Dateigrößen der einzelnen analysierten Komponenten.

Bezeichnung	**Version**	**MD5-Prüfsumme**	**Größe in Bytes**
Dropper	*BckR2D2-II*	`309ede406988486bf81e603c514b4b82`	643.072
Softwarebibliothek	*BckR2D2-I*	`b5080ea3c9a25f2ebe0fb5431af80c34`	364.544
	BckR2D2-I'	`930712416770a8d5e6951f3e38548691`	360.448
	BckR2D2-II	`934b696cc17a1efc102c0d5633768ca2`	356.352
Kernel-Level-Treiber (32-bit)	*BckR2D2-I*	`d6791f5aa6239d143a22b2a15f627e72`	5.376
	BckR2D2-I'	`d6791f5aa6239d143a22b2a15f627e72`	5.376
	BckR2D2-II	`9a8004e2f0093e3fe542fa53bd6ad1b2`	10.112
Kernel-Level-Treiber (64-bit)	*BckR2D2-II*	`cd01256f3051e6465b817fffc97767dd`	262.730
User-Level-Anwendung	*BckR2D2-II*	`976dd8be30df45c6fb2b4aaaa9ce9106`	155.648
Remover	*BckR2D2-II*	`96c56885d0c87b41d0a657a8789779f2`	40.960

Tabelle 3: Komponenten der untersuchten Schadsoftware mit ihren zugehörigen Prüfsummen und Dateigrößen