Spionage via Webcam: Welchen Schutz bieten Personal Firewalls und Virenscanner

Ulrich Greveler[1] · Matthias Wellmeyer[1]

Fachhochschule Münster[1]
Labor für IT-Sicherheit
{greveler | wellmeyer}@fh-muenster.de

Zusammenfassung

Motiviert durch die spektakulären Fälle aus den Jahren 2010 und 2011, die tiefgreifende Verletzungen der Privatsphäre von Nutzern durch heimliche Beobachtung mit der Webcam zum Gegenstand hatten, beschreibt der Beitrag den Aufbau und die Analyse einer Webcam-Spionagesoftware.
Neben Industriespionage stellen Voyeurismus und heimliche Überwachung die Motivation für diese Rechtsverletzungen dar. Bei der untersuchten Software wird die Multimediafähigkeit moderner Computer ausgenutzt, um Benutzer, während diese am Rechner (z. B. Notebook) arbeiten, und die Umgebung über die Webcam unbemerkt zu beobachten. Zu diesem Zweck wurde in einer Laborumgebung ein Demonstrator entwickelt, der in einem festgelegten Zeitintervall das Bild der Webcam zu einem Angreifer überträgt. Der Prototyp demonstriert, dass Firewall und Virenscanner zwar technische Hürden darstellen; diese sind aber – so zeigen es die technischen Laborergebnisse – überwindbar und täuschen dem Benutzer daher einen Schutz vor, der nicht existiert.

1 Einführung

1.1 Heimliche Beobachtung mit Webcam

In jüngster Vergangenheit wurden spektakuläre Fälle bekannt, die tief greifende Verletzungen der Privatsphäre von Nutzern durch heimliche Beobachtung mit der Webcam zum Gegenstand hatten: Im Juli 2010 wurde ein Fall bekannt [WDR10], nachdem die Staatsanwaltschaft Aachen die Wohnung eines mutmaßlichen Täters durchsuchte, der über hundert Mädchen in ihren Kinderzimmern unter Nutzung einer Spionagesoftware beobachtet haben soll. Im Oktober 2010 wurde berichtet, dass ein Student der Rutgers University (New Brunswick, USA) sich das Leben genommen hatte, nachdem er via Webcam bei sexuellen Handlungen gefilmt worden war und das Videomaterial im Internet verbreitet wurde [FoxN10].

Diese Vorfälle, sowie die Tatsache, dass Industriespionage über mit dem PC verbundene Kameras beobachtet wird [BR10], sorgen dafür, dass der softwareseitige Zugriff auf Webcams verstärkt in den Fokus von Sicherheitsbetrachtungen rückt. Ein Bericht des IT-Security-Magazins *Securitymanager.de* berichtet, dass die Anzahl erstmalig in Erscheinung tretender Schadprogramme seit dem Jahr 2001 exponentiell wächst [Mash07], was die Gefahr eines Angriffes auch für Privatpersonen erhöht. Ein weiteres Risiko ist die steigende Anzahl von Multimedia-PCs (z. T. mit Webcams) in Haushalten. Eine Statistik des *Bundesverbands Informationswirtschaft, Telekommunikation und neue Medien e.V.* zeigt, dass bereits 82% der

Haushalte in Deutschland einen Computer besitzen [Bitk10] und damit potentielle Angriffsziele sind.

Zunächst wird ein Überblick über üblicherweise installierte Sicherheitssoftware wie (sogenannte) „Personal Firewalls“ und Virenscanner, sowie verschiedene Scan-Techniken der Virenscanner und die grobe Funktionsweise einer Firewall gegeben. Anhand dieser Ausgangsbasis wird deutlich, warum es weiterhin eine Herausforderung darstellt, eine Schadsoftware von ungefährlicher Software zu unterscheiden.

Da Microsoft-Betriebssysteme den höchsten Marktanteil besitzen, wird der hier beschriebene Demonstrator (Prototyp einer Spionagesoftware) dieser Betriebssystemlandschaft angepasst. Nach einer Statistik von *Net Applications* ist das Produkt Windows-XP mit 52.41% immer noch das meist genutzte Betriebssystem [Appl11] und wird daher in der Laborumgebung verwendet. Wir geben jedoch im Abschnitt 4 Hinweise zur Übertragbarkeit der Erkenntnisse auf die Produkte *Windows Vista* und *Windows 7*.

1.2 Sicherheitssoftware

Ein Grund für die Schwierigkeiten bei der Erkennung der Schadsoftware ist die Funktionsweise der Sicherheitstools. Daher wird hier ein kurzer Überblick in die Funktionsweise von Virenscanner und Firewall gegeben.

Virenscanner

Virenscanner (auch als Anti-Virus- oder kurz AV-Software bekannt), dienen der Erkennung und Entfernung bekannter Schadsoftware. Virenscanner lassen sich primär in *On-Demand*- und *On-Access*-Virenscanner unterteilen. *On-Demand-Virenscanner* werden nur nach Aufforderung durch den Benutzer und/oder einem zuvor festgelegtem Zeitpunkt aktiv. Der *On-Access-Virenscanner* hingegen überprüft vor jedem Zugriff die entsprechende Datei. Um ein möglichst hohes Maß an Sicherheit zu erlangen, verwenden die meisten Virenscanner sowohl die *On-Demand* als auch die *On-Access* Technologie.

Auch bei der Erkennung von Schadsoftware lassen sich Virenscanner in zwei Kategorien unterteilen. Die klassische *reaktive* (oder auch *signaturbasierte*) Erkennung ist sehr effektiv gegen schon bekannte Schadsoftware. Durch eine bekannte Byte folge in der Schadsoftware lässt sich diese sehr schnell identifizieren. Der Virenscanner muss dafür nur den Binärcode des Programms nach ihm bekannten Signaturen (aus seiner Signaturdatenbank) durchsuchen. Für eine erfolgreiche Identifizierung bekannter Schadsoftware stellen die Hersteller regelmäßige Updates der Signaturen von neu aufgetretener Schadsoftware bereit. Der Nachteil dieses Verfahrens ist, dass Hersteller der Entwicklung zeitlich hinterherhinken, da neue Schadsoftware zuerst als solche erkannt und im Anschluss die Signaturdatenbank aktualisiert werden muss.

Eine Vermeidung der zeitlichen Verzögerung bis zur Aktualisierung der Signaturdatenbank bieten *proaktive* (oder *heuristische*) Verfahren, die auf Grund bestimmter Merkmale einer Anwendung Schadsoftware identifiziert. Aufgrund der hohen Komplexität zur Erkennung ist eine Unterscheidung sehr aufwendig und führt je nach Empfindlichkeit entweder zu einer hohen Anzahl an falschen Meldungen oder aber zu einer Nichtidentifizierung von Schadsoftware.

Neben dem *On-Demand*- und *On-Access*- Virenscanner gibt es noch folgende Scanverfahren:

Sandboxing: Bei dem *Sandboxing* Verfahren wird in einer virtualisierten Umgebung der Programmablauf Schritt für Schritt analysiert. Dabei wird Hardware und Software des Sy-

stems emuliert und es kann am Ende eine detaillierte Aussage über den Effekt des Programms gegeben werden.

Online-Scanner: Viele Antivirus-Hersteller bieten auch *Online-Scanner* an. Neben einem vollständigen Scan des Systems bieten diese meist auch die Möglichkeit, einzelne Dateien online zu analysieren.

Cloud-Antivirus: Die Signaturen von Schadsoftware liegen hierbei „in der Cloud“. Die Reaktionszeit bei neuer Schadsoftware soll hierbei minimiert werden. Ist die Signatur neuer Schadsoftware bekannt, kann diese ohne Verzug über die Cloud zur Verfügung gestellt werden.

Personal Firewall

Eine Personal Firewall (auch Desktop-Firewall genannt) zielt auf den Markt für private Benutzer. Das System, auf dem diese Firewall installiert ist, soll vor unbeabsichtigten Netzwerkzugriffen geschützt werden. Personal Firewalls weisen Nachteile auf, die in der Architektur und in der Konfiguration liegen: Die Architektur sieht vor, dass die Personal Firewall eine Software ist, die auf dem Betriebssystem operiert. Wurde das Betriebssystem kompromittiert, kann ein Angreifer die Firewall manipulieren oder sogar abschalten, um unbemerkt Daten aus dem System zu schleusen. Problematisch ist zudem die Bedienung der Software durch den Endbenutzer selbst, der Entscheidungen über die Zulässigkeit von Netzwerkverbindungen treffen muss, ohne über ausreichende Informationen bzw. benötigte Fachkenntnisse zu verfügen. Die Hersteller bieten dem Benutzer ein grafisches Frontend für die Konfiguration der Firewall an, allerdings bleibt dabei unklar, wie das System sicher zu konfigurieren ist, ohne die Benutzbarkeit wesentlich einzuschränken.

Das Bundesamt für Sicherheit in der Informationstechnik (BSI) empfiehlt eine Personal Firewall eingeschränkt als Ergänzung zu einer zentralen (Netzwerk-) Firewall, die das gesamte Netzwerk zusätzlich absichert. Zudem sollen Protokolldaten regelmäßig durch fachkundiges Personal ausgewertet werden. [BSI05] Ein derartiger Einsatz in Privathaushalten dürfte allerdings Seltenheit haben und löst zudem das Problem mangelnden Fachwissens des Endbenutzers bei der initialen Konfiguration nicht.

2 Technische Details

Im Folgenden werden technische Details sowohl zur Testumgebung als auch zur technischen Umsetzung des Prototypen beschrieben. Außerdem wird nach einer Angriffsstelle gesucht, die sich vom Prototyp ausnutzen lässt und es ermöglicht, das durch den Prototypen erfasste Bild unbemerkt zu übertragen. Im Anschluss werden jeweils drei Sicherheitstools (drei Personal Firewalls und drei Virenscanner) ausgewählt, um zu überprüfen, ob diese einen Schutz gegen die entwickelte Spyware bieten. Abschließend wird der modifizierte Demonstrator über Online-Virenscanner mit rund 45 verschiedenen Virenscannern überprüft.

2.1 Betriebssystem

Wie in der Einführung beschrieben, wird ein Testsystem, basierend auf dem Produkt *Microsoft Windows XP Professional* (inclusive Service Pack 3) und dem *.NET-Framework 3.5* (inclusive Service Pack 1) verwendet. Das .NET-Framework spielt dabei eine wichtige Rolle, da es eine

Sammlung von Klassenbibliotheken, Programmierschnittstellen und Dienstprogrammen enthält und damit als „Unterbau“ für den Webcam-Zugriff benötigt wird.

2.2 Testumgebung

Um später jede einzelne Softwareversion (sowohl Personal Firewall als auch Virenscanner) separat und unter gleichen Umständen testen zu können, wurde das System als Referenzinstallation vorgenommen und als Image gesichert. Durch eine automatisierte Rückspielung des Images wird damit ein nach strengen Maßstäben identischer Ausgangszustand erreicht. Da an die Konfiguration des Empfängers keine Bedingungen bezüglich des Ausgangszustandes geknüpft sind, wird dieser einmal eingerichtet und während der gesamten Tests nicht verändert.

2.3 Webcam-Zugriff

Zur Untersuchung der Sicherheit wurde im Rahmen einer Bachelorarbeit [Well10] ein Prototyp entwickelt, der eine Schnittstelle des Betriebssystems nutzt, um das Signal der Webcam abzufangen und dieses einem Angreifer zu übermitteln.

Um Zugriff auf die Webcam zu erlangen, bietet Microsoft seit der Version *Windows Server 2003 SP1* (das Service Pack 1 wurde am 30. März 2005 veröffentlicht) die *DirectShow*-Architektur an. Diese war ursprünglich ein Teil des *DirectX SDK*, wurde aber seit Erscheinen von Windows Server 2003 SP1 in das *Windows SDK* verschoben und ist technisch gesehen damit ein Teil der Windows-Plattform geworden [Micr]. Die API wurde von Microsoft für Softwareentwickler bereitgestellt, um verschiedenste Operationen mit Media-Dateien oder Streams zu verarbeiten.

DirectShow besteht aus einer modularen Architektur, die es erlaubt, verschiedene Komponenten zu einem Filtergraph zusammenzufügen. Die Komponenten werden als *Filter* bezeichnet und in verschiedene Kategorien aufgeteilt. Für den Zugriff auf eine Webcam sind lediglich drei Filter von Bedeutung:

Source Filter: Der *Source Filter* ist der Eintrittspunkt in den Graph. Die Daten können dabei von einer Datei, einer Kamera oder direkt aus dem Netzwerk kommen. Für jede Datenquelle ist jeweils ein entsprechender Filter zuständig.

Transform Filter: Ein *Transform Filter* erhält einen Stream, verarbeitet die Daten und gibt den Stream weiter.

Renderer Filter: Am Ende des Filtergraphen ist der *Renderer Filter*, der Daten empfängt und diese dem Benutzer präsentiert.

Ein Filtergraph der, wie durch diese Anwendung benötigt, Signale einer Video- oder Audio-Quelle erfasst, wird *Capture-Graph* genannt. Für den Zugriff auf die Webcam ist demnach ein einfacher Capture-Graph mit mindestens drei Komponenten nötig:

Source Filter zum Einführen des Signals der Webcam in den Graph.

Transform Filter zum Entnehmen eines Bildes aus dem Datenstream.

Renderer Filter zum Darstellen des Streams.

Da das Bild der Webcam in diesem Fall nicht dargestellt werden soll, wird als *Renderer Filter* ein „Nullrenderer“ eingefügt, der den Stream verwirft, anstatt ihn dem Benutzer zu präsentieren.

Als Datenübertragung wird ein einfaches *Send-and-Acknowledgement*-Konzept genutzt. Der Sender zerteilt dazu die Daten in kleine Teile, die jeweils in einem UDP-Paket an den Empfänger

geschickt werden. Da das UDP-Protokol ein minimales verbindungsloses Netzwerkprotokoll ist, ist nicht sicher gestellt, dass alle Daten, die gesendet wurden, auch bei dem Empfänger ankommen. Aus diesem Grund bestätigt der Empfänger jedes einzelne Paket. Durch die Nutzung des UDP-Protokolles ist die Implementierung wegen der nicht zugesicherten Übertragung zwar aufwendiger, dafür aber auch unauffälliger (es existiert keine stehende Netzwerkverbindung, die dem Benutzer angezeigt werden könnte).

Wie der Großteil der Betriebssysteme bietet auch Microsoft eine Socket-API für den Zugriff auf den Netzwerksocket an. Für die Datenübertragung sind daher nur die drei Schritte

- Initialisierung der Netzwerkverbindung und des Sockets
- Übertragung des abgefangenen Bildes der Webcam
- Beenden der Nutzung von Windows-Sockets

notwendig.

Nach erfolgreicher Implementierung der Spyware muss nun ein Weg gefunden werden, um die (üblicherweise) vorhandene Personal Firewall und den Virenscanner zu umgehen.

2.4 Ausgenutzte Schwachstelle

Nach Installation und Konfiguration der ersten Sicherheitstools (zunächst getestet wurden *Zonealarm Firewall* und *Avant Antivir*) wurde das System genauer analysiert. Dabei ist der Prozess *svchost.exe* in den Blickpunkt geraten, da er eine besondere Rolle spielt. Wie in Abbildung 1 zu

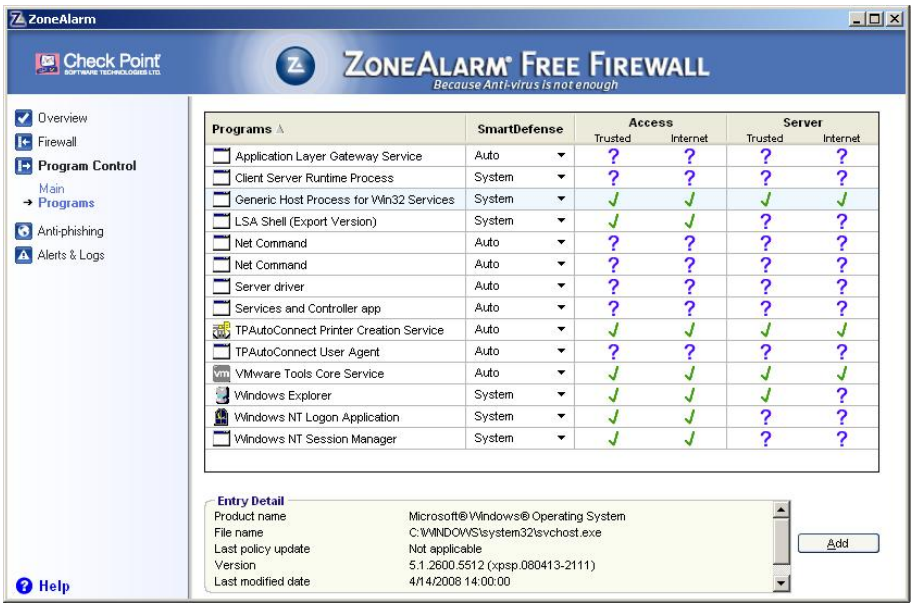

Abb. 1: ZoneAlarm default Konfiguration

sehen ist, hat der *Generic Host Process for Win32 Services* vollen Zugriff auf das Internet. Die ausführbare Datei *„svchost.exe“* befindet sich unter *„%SystemRoot%\System32“* und hat nach Herstellerangaben folgende Funktion:

> Beim Start überprüft „Svchost.exe“ den auf Dienste bezogenen Abschnitt der Registrierung, um eine Liste von Diensten zusammenzustellen, die geladen werden müssen. Es können gleichzeitig mehrere Instanzen von „Svchost.exe“ ausgeführt

> werden. Jede Svchost.exe-Sitzung kann eine eigene Gruppe von Diensten enthalten. Es hängt also von der Art und Weise des Svchost.exe-Aufrufs ab, welche verschiedenen Dienste ausgeführt werden. [Supp10]

Die verschiedenen Gruppen sind in der Registrierung unter dem Schlüssel `HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\Svchost` zu finden. Unter diesem Schlüssel befinden sich mehrere Werte, die jeweils eine svchost-Gruppe bilden. Ist ein Dienst einer Gruppe aktiv, so wird dieser in Form einer svchost.exe-Instanz angezeigt. Da der betriebssystemeigene *Taskmanager* keinen Aufschluss über die aktiven Dienste einer svchost.exe liefert, ist dem Benutzer regelmäßig nicht bekannt, welche Funktionen sich hinter dieser verbergen. Um dennoch die Funktionalität der derzeit aktiven svchost-Instanzen zu erhalten, gibt es das Konsolentool *tasklist*. Das Tool bietet durch die Nutzung des „`/SVC`"-Parameters die Möglichkeit, die Dienste eines Prozesses aufzulisten. Mit Hilfe des Filter Parameters „`/FI`" kann dann zusätzlich noch auf den Prozess svchost.exe gefiltert werden. In der Ausgabe (siehe Abbildung 2) ist dann zu sehen, dass sechs svchost.exe-Instanzen gestartet sind,

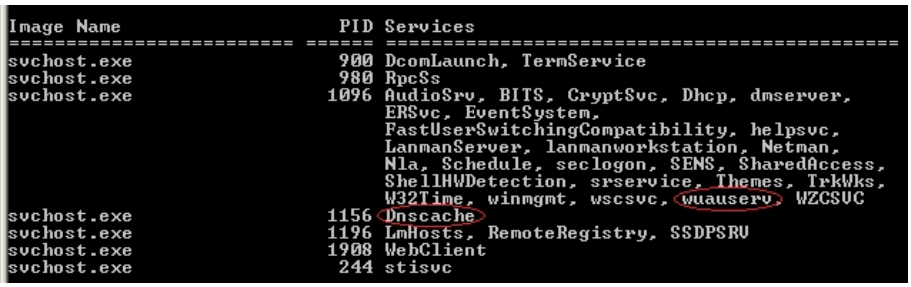

Abb. 2: Dienste der *svchost.exe* Instanzen

in denen neben anderen die Dienste *wuauserv* und *Dnscache* aktiv sind. Der Dienst *wuauserv* aktiviert den Download und die Installation von automatischen Updates des Betriebssystems und der *Dnscache* löst Domain-Name-System-Namen (DNS-Anfragen) für den Computer auf und speichert diese zwischen (Caching). Beide Dienste benötigen einen Zugriff auf das Internet und sind daher in den Personal Firewalls als erlaubte Netzwerkaktivität einzutragen.

2.5 Virenscanner

Die hier getesteten Virenscanner

- Avast - Free Antivirus 5.0
- Avira - AntiVir Personal (FREE Antivirus) 10.0
- AVG - Anti-Virus Free 9.0

sind nach einer Studie [Grou10] der *OPSWAT Group* die drei am meisten verbreitetsten Virenscanner weltweit. Um sicherzustellen, dass die Virenscanner funktionsfähig sind, wurden diese nach erfolgreicher Installation und Online-Update in der Laborumgebung mit Hilfe einer Testdatei überprüft. Diese Datei enthält einen vom „European Institute for Computer Antivirus Research" (kurz: *EICAR*) entwickelten Teststring. Ist der Virenscanner funktionsfähig, wird die Datei als „Testvirus" erkannt [EICA].

Der Test der entwickelten Spyware zeigte, dass die (oft angepriesene) Malware-Erkennung der Antivirussoftware für diesen Angriffstyp (Ausspähen mit Hilfe der Webcam) keinen Sicherheitsvorteil darstellt. Der abschließende Test mit den zwei Online-Virenscannern *Virustotal*

[Sist] und *Jotti's malware scan* [Jott11] ergab keinerlei Positiv-Ergebnisse. Der modifizierte Demonstrator wird also von keinem der 45 Virenscannern erkannt.

Aufgrund der individuellen Struktur der Spyware ist den Herstellern von Antivirus-Software die Signatur des Prototypen nicht bekannt, was ein Erkennen allein durch die Signaturdatenbank offenbar unmöglich macht. Da der Prototyp jedoch eine Netzwerkverbindung nach außen öffnet und Zugriff auf die Ressourcen Webcam vornimmt, wäre eine Erkennung durch das *proaktive* Scanverfahren möglich. Es zeigt sich jedoch, dass die derzeit im Markt verbreitete Antivirussoftware dieses nicht leistet und dem Nutzer keinerlei Schutz bietet. Eine Herausforderung auf Seiten der AV-Hersteller ist die Problematik zur Unterscheidung der Spyware von einer Videochat/Videokonferenz-Software wie Skype oder ICQ.

Das einzige verbleibende Erkennungsmerkmal für den Nutzer ist die oft vorhandene LED, die Kameraaktivitäten anzeigt. Leider gibt es bezüglich dieser Anzeige keine Standards, die festlegen, ob diese überhaupt vorhanden sein muss und ob sie softwareseitig deaktiviert werden kann. So hat beispielsweise das multimediafähige *Asus-X55S*-Notebook keine LED, welche die Kameraaktivität anzeigen könnte. Darüber hinaus zeigen die öffentlich gewordenen Fälle, dass ein Dauerleuchten der LED für den Nutzer nicht selten unbemerkt bleibt oder von ihm nicht als Hinweis für Spionageaktivitäten interpretiert wird.

2.6 Personal Firewall

Die drei getesteten Personal Firewalls sind:

- ZoneAlarm - Free Firewall 9.2
- Ashampoo - FireWall FREE 1.20
- Sygate - Personal Firewall 5.6

Wird der Prototyp ohne weitere Anpassung gestartet, erkennen die Personal Firewalls die Anwendung und beginnen eine Interaktion mit dem Benutzer. Dieser muss dann nach eigenem Ermessen entscheiden, ob er der Anwendung Zugriff auf das Internet gewähren möchte oder nicht. Wird jetzt allerdings die *svchost.exe* unter *„%SystemRoot%\System32“* durch den entwickelten Prototypen ersetzt und gestartet, aktzeptieren *ZoneAlarm* und *Ashampoo* ohne Warnung oder Nachfrage einen Verbindungsaufbau nach außen! Lediglich die *Sygate*-Firewall gibt eine Warnung über den versuchten Verbindungsaufbau aus. Für den Benutzer ergibt sich aber auch bei dieser Warnung die Schwierigkeit, dass sich die erzeugte Mitteilung kaum von einer Warnung unterscheidet, die bei regelmäßigen automatischen Online-Updates des Betriebssystems erscheint. Der Benutzer müsste sich den Hinweis des Firewall-GUIs im Detail durchlesen und entsprechende Fachkenntnisse über betriebssysteminterne Vorgänge haben, um zu bemerken, dass es sich hierbei **nicht** um ein automatisches Update durch den *wuauserv*-Dienst sondern um eine Fremdsoftware handelt.

Als noch fataler stellt sich die Situation dar, wenn der Nutzer, der sich von den regelmäßigen Warnungen nach Systemupdates belästigt fühlt, den Zugriff für die *svchost.exe* auf „immer akzeptieren“ stellt. In dem Fall gibt es keine Warnung der Firewall, und der Nutzer hat ohne weitere Werkzeuge keine Möglichkeit mehr, das Versenden des Kamerabildes zu bemerken. Auffällig ist zudem, dass für die Firewall offenbar nur der Pfad und der Dateinamen ausschlaggebend ist. Sobald eine *svchost.exe* im Verzeichnis *„%SystemRoot%\System32“* aktiv ist, unterliegt diese automatisch der Firewall-Regel für die *svchost.exe*. Im Labortest ist hier insbesondere die Ashampoo-Firewall negativ aufgefallen, da diese nicht einmal die Möglichkeit bietet,

den Netzwerkzugriff für die *svchost.exe* zu sperren. Es existieren „Interne Regeln“, die weder eine Änderung, noch die Löschung dieser zulassen (siehe Abbildung 3). Das Produkt *Zone-*

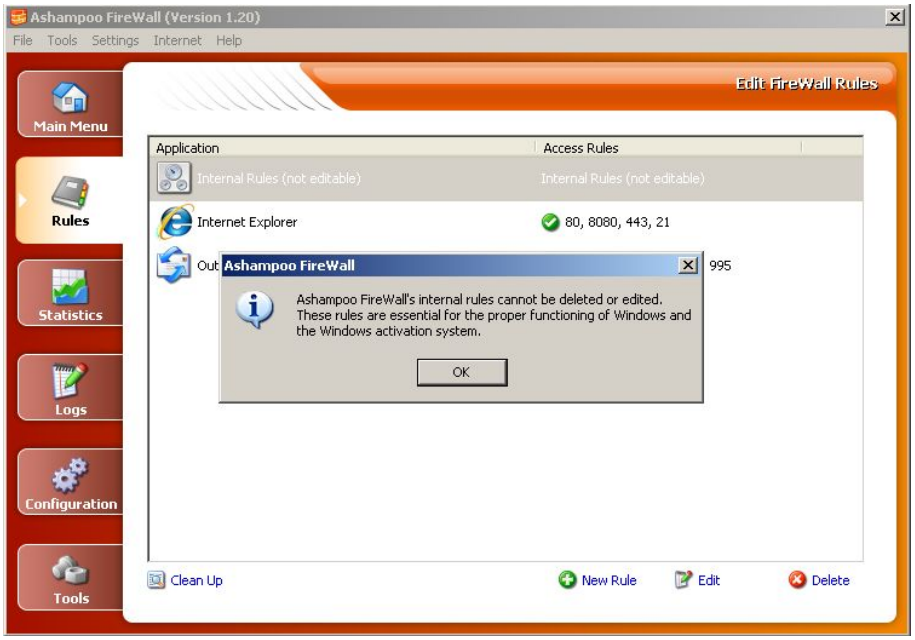

Abb. 3: Die Ashampoo Firewall verweigert jedliche Änderung der „Internen Regeln“

Alarm dagegen erlaubt zwar in der Standardkonfiguration den Netzwerkzugriff, bietet aber die Möglichkeit, den Zugriff manuell zu sperren (vgl. Abbildung 1).

3 Modifizierung des Prototypen

Um den Prototypen nun so zu modifizieren, dass dieser sich automatisch als *svchost.exe* ausgibt, sind zwei Schritte nötig. Mit Hilfe eines HEX-Editors wird der Binärcode des Prototypen extrahiert und als Konstante in ein zweites Programm eingefügt. Da die *svchost.exe* sich nicht direkt löschen oder überschreiben lässt, muss das zweite Programm die Datei zuerst umbenennen und im Anschluss daran eine eigene *svchost.exe* mit dem extrahiertem Binärcode erstellen und ausführen. Ein verbleibender Hinweis auf die gefälschte *svchost.exe* ist für einen technisch versierten Benutzer im Taskmanager zu sehen, da das Programm ungewöhnlicherweise in der Rolle des angemeldeten Benutzers ausgeführt wurde (siehe Abbildung 4). Durch eine Eigenheit des Betriebssystems wird jeglicher Hinweis auf die falsche *svchost.exe*, im Dateisystem, vom Betriebssystem verwischt. Die Datei wird vom Betriebssystem *Windows XP* periodisch neu geschrieben (!) und beseitigt dabei automatisch die Spuren der Spyware. Da die Software vollständig im Speicher ausgeführt wird, gibt es keinerlei Laufzeitfehler durch das Überschreiben. Technische Hintergründe für das überraschende Verhalten des Betriebssystems konnten durch die Autoren nicht ermittelt werden.

3.1 Windows Vista und Windows 7

Generell ist der unveränderte Prototyp auch auf den Produkten *Windows Vista* und *Windows 7* lauffähig. Problematisch bei der modifizierten Version des Prototypen ist die neue Benutzerkontensteuerung (UAC - **U**ser **A**ccount **C**ontrol). Diese verhindert unter anderem einen Zugriff auf Dateien im *„%SystemRoot%\System32“* Verzeichnis. Es ist zwar möglich, Zugriff auf die

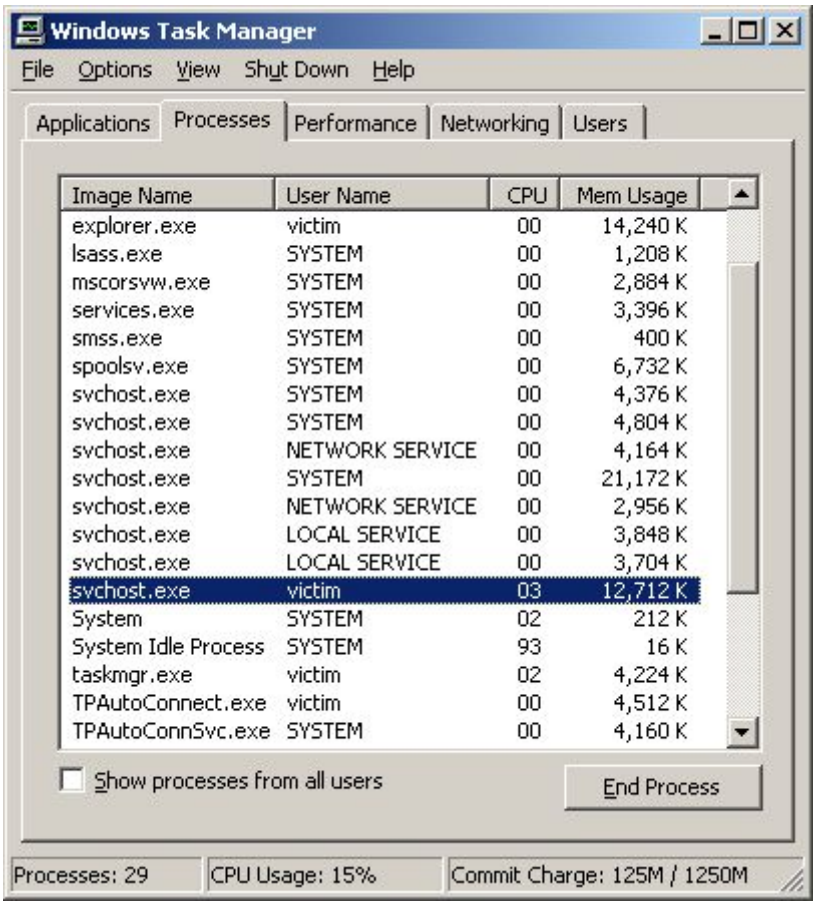

Abb. 4: Taskmanager mit zusätzlicher *svchost.exe* Instanz

Dateien zu erlangen, allerdings müssen dazu erst die Berechtigungen angepasst werden. Da diese Änderungen von der Spyware nicht ohne erheblichen Aufwand bzw. ohne erfolgreiche Täuschung des Benutzers vorgenommen werden können, stellt diese neue Benutzerkontensteuerung einen gewissen Sicherheitsvorteil dar. Es wäre in einer zukünftigen Untersuchung zu klären, ob die Anpassung einer Berechtigung ein geeignetes Merkmal darstellt, das die heuristische Erkennung von bisher unbekannter Malware ermöglicht.

4 Mögliche Gegenmaßnahmen

Aufgrund der neuen Benutzerkontensteuerung seit Windows Vista ist es für die Spyware nicht mehr ohne zusätzliche Privilegien möglich, Dateien im *„%SystemRoot%\System32“* zu verändern. Änderungen der Anwendungsprivilegien, insbesondere wenn sie Anwendungen betreffen, die nach außen kommunizieren, können als heuristisches Erkennungsmerkmal herangezogen werden; hierbei wird ein Grenzbereich betreten, den sowohl AV- als auch Firewall-Hersteller besetzen könnten.

Eine andere Maßnahme ist der kombinierte Schutz durch Sicherheitssoftware. Wie im Kapitel 2.5 dargestellt, ist es für Antivirus-Software ein bisher ungelöstes Problem, diese Art von Schadsoftware zu erkennen. Es wäre jedoch nahe liegend, bei Zugriff auf das *System32*-Verzeichnis heuristisch Alarm zu schlagen und eine Warnung an den Benutzer zu senden, insbesondere dann, wenn dieselbe Software auch auf die sensible Ressource Webcam zugreift. Das

heuristische Erkennungsmerkmal wäre dabei nicht eine der beiden einzelnen Aktionen (Zugriff auf *System32*-Verzeichnis und *Webcam*), sondern die Kombination von beiden Merkmalen.

Auf Seiten der Firewall-Hersteller gäbe es die Option, nicht nur die Quelle der Anfrage sondern auch ein Erkennungsmerkmal der Anwendung zu überprüfen. Aufgrund der Tatsache, dass sich die Prüfsumme von systemnahen Anwendungen außerhalb der regelmäßigen Updates nicht ändert, wäre dies ein Mechanismus, die Malware-Anwendung zu identifizieren. Eine (scheinbare) System-Applikation, die Netzwerkverbindungen aufbaut und eine geänderte Prüfsumme besitzt, kann dem Benutzer vorsorglich gemeldet werden. Eine Sicherheitspolitik dieser Art könnte jedoch eine Warnung nach fast jedem Softwareupdate bewirken, sodass ein erhöhtes Maß an Sicherheit mit einer verringerten Benutzerfreundlichkeit erkauft wird.

Eine herstellerübergreifende Inititive mit dem Ziel, die Spionage über die Webcam zu erschweren, könnte die Standardisierung der Webcam-LED vornehmen. Da einige Kameras gänzlich ohne LEDs operieren, aber auch ein konstant leuchtendes Licht nicht sehr auffällig ist, würde beispielsweise eine standardisierte, blinkende LED absehbar mehr Aufmerksamkeit auf sich ziehen. Da die Webcam aus Sicherheitssicht nur durch den Benutzer bewusst aktiviert werden sollte, würde das Blinken während der Nutzungszeit nur unerheblich stören, dafür aber zur Aufmerksamkeitssteigerung bei fremdgesteuerter Kameranutzung beitragen. Eine Abschaltung des Blinkens für „Dauernutzer“, die sich gestört fühlen, könnte über einen geschützen Zugriff auf das BIOS ermöglicht werden, ohne die sichere Standardkonfiguration zu gefährden. Schließlich wäre auch in Beracht zu ziehen, eine manuelle Öffnung einer Abdeckung über der Kameralinse vorzusehen: Ein Nutzer, der die Kamera nicht benötigt, kann diese Abdeckung geschlossen lassen und ist damit vor dem Ausspähen geschützt.

5 Zusammenfassung

Zusammenfassend stellen wir fest, dass weiterhin eine erhebliche Überwachungslücke gegenüber der heimlichen Beobachtung mit Webcams besteht.

Eine Spionagesoftware gerichtet auf ein konkretes Zielsystem (beispielsweise für den einmaligen Einsatz), kann mit geringem technischen Aufwand erstellt werden; eine Erkennung durch Virenscanner und Personal Firewalls kann der Angreifer in Bezug auf die von uns getestete Systemlandschaft wirksam ausschließen. Es genügt, wenn das Opfer die Applikation, die es vorher mit der AV-Software gescannt hat, einmalig ausführt.

Die heuristische Erkennung einer Malware, die eine Spionage über Webcam ermöglicht, würde den Schutz der Besitzer eines mit einer Kamera ausgestatteten Systems deutlich erhöhen; es ergeben sich gleich mehrere heuristische Merkmale, die in Kombination genutzt werden können: Eine Heuristik wäre somit implementierbar. Den Autoren dieses Beitrags sind die Gründe nicht bekannt, die die Hersteller von Antivirus- und Personal-Firewall-Software derzeit hindern, einen Schutz gegen Kamera-Spionage in die Funktionalität ihrer Produkte aufzunehmen. Es liegt jedoch die Vermutung nahe, dass der hier betroffene Grenzbereich zwischen Softwareanalyse und Netzwerküberwachung, der ein Zusammenwirken von verschiedenen Sicherheitsprodukten erforderlich macht, nur zögernd betreten wird: Antivirus-Softwarehersteller und Firewall-Hersteller müssten sich hier auf ein Vorgehen einigen, um eine Ressourcenüberwachung zu ermöglichen. Ein überlappender Überwachungsbereich führte schnell zu technischen Konflikten, die in für den Benutzer unangenehmen Fehlermeldungen mündeten; es liegt daher nahe, den jeweils angestammten Bereich nicht zu verlassen: Eine gravierende Sicherheitslücke ist jedoch die nicht akzeptable Folge.

Literatur

[Appl11] N. Applications: Global Market Share Statistics. http://marketshare.hitslink.com/Default.aspx (2011).

[Bitk10] Bitkom: Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V.: ITK-Infrastruktur. http://www.bitkom.org/de/markt_statistik/46261.aspx (2010).

[BR10] BR: Bayerischer Rundfunk: Bericht vom 11.01.2010: Notizbuch zum Thema Datenklau (2010).

[BSI05] BSI: M 5.91 Einsatz von Personal Firewalls für Internet-PCs. www.bsi.bund.de/cln_165/ContentBSI/grundschutz/kataloge/m/m05/m05091.html (2005).

[EICA] EICAR: European Institute for Computer Antivirus Research: The Anti-Virus or Anti-Malware test file.

[FoxN10] Fox-News: Rutgers president defends response in suicide case. http://www.foxnews.com/us/2010/10/07/rutgers-president-defends-response-suicide-case/ (2010).

[Grou10] O. Group: Windows Antivirus - Worldwide Market Share Analysis. http://www.oesisok.com/news-resources/reports/av_usage.pdf (2010).

[Jott11] Jotti: Jottis Malwarescanner. http://virusscan.jotti.org (2004 - 2011).

[Mash07] Y. Mashevsky: Kampf im Cyberspace - wer wird gewinnen? Securitymanager.de (2007).

[Micr] Microsoft-Corporation: DirectShow. http://msdn.microsoft.com/en-us/library/ms783323(VS.85).aspx.

[Sist] H. Sistemas: VirusTotal. http://www.virustotal.com.

[Supp10] M. Support: Beschreibung des in Windows XP Professional Edition enthaltenen Programms SSvchost.exe". http://support.microsoft.com/kb/314056/de (2010).

[WDR10] WDR: Westdeutscher Rundfunk: Bericht vom 16.07.2010: Mädchen übers Internet beobachtet (2010).

[Well10] M. Wellmeyer: Spyware development and analysis (Bachelorarbeit). Fachhochschule Münster, Labor für IT-Sicherheit (2010).