VU VRIJE UNIVERSITEIT AMSTERDAM

VU Research Portal

Keylogger Detection and Containment

Ortolani, S.

2013

document version
Publisher's PDF, also known as Version of record

Link to publication in VU Research Portal

citation for published version (APA)
Ortolani, S. (2013). *Keylogger Detection and Containment*. [PhD-Thesis - Research and graduation internal, Vrije Universiteit Amsterdam].

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

E-mail address:
vuresearchportal.ub@vu.nl

Download date: 14. Dec. 2024

Samenvatting

Onze recente geschiedenis heeft vele schrijvers geïnspireerd tot het schrijven van spionageverhalen waarin twee of meer spionnen het delicate en gevaarlijke spel van het stelen van privé-informatie speelden. Een essentiële vaardigheid hierbij was om telefoons en telexen af te luisteren zonder dat de andere partij het merkte. Meestal vereiste dit precieze installatie van kleine apparatuur direct in de communicatie hardware. Met de opkomst van computers werden deze kleine apparaatjes tot wat we nu kennen, namelijk hardware keyloggers: kleine dongles geplaatst tussen het toetsenbord en het moederbord, ontworpen om alle toetsaanslagen van de gebruikers op te slaan. Ook in het geval van het installeren van hardware keyloggers, was echter fysieke toegang nodig. Hierdoor kon men (wij) zich nog relatief veilig voelen, omdat wij ervan uitgaan dat het in onze macht ligt om te voorkomen dat een inbreker inbreekt. Sommige mensen voorzien hun appartement van een alarmsysteem, anderen houden een pistool onder hun kussen.

Software keyloggers zijn de "massamarkt" versie van deze hardware-apparaten. Eenmaal geïnstalleerd op de computers van gebruikers, verlenen zij toezicht op de activiteit van de gebruiker door op slinkse wijze alle toetsaanslagen te loggen en, in sommige gevallen, door te spelen aan derden [36]. Als ze ontworpen zijn om te worden uitgevoerd in user-space zijn ze ook eenvoudig te implementeren: alle moderne besturingssystemen bieden ongeprivilegieerde Application Programming Interfaces (APIs), die kunnen worden ingezet voor het onderscheppen van alle toetsaanslagen van de gebruikers. In tegenstelling tot keyloggers uitgevoerd als kernel-modules, is er geen toestemming nodig voor de installatie en het uitvoeren van deze keyloggers. Een gebruiker kan de keylogger ten onrechte beschouwen als een onschuldig stukje software en op deze manier worden misleid tot de uitvoering ervan. Kernel keyloggers

vereisen—naast afhankelijkheid van toestemming van de gebruiker voor zowel de uitvoering als de installatie—, dat de programmeur afgaat op kernel-level facilititeiten om alle berichten die zijn verzonden door de toetsenbord-driver, te kunnen onderscheppen. Het vergt zeker een aanzienlijke inspanning en kennis voor een effectieve en bug-vrije implementatie hiervan.

In het licht van deze observaties, is het geen verrassing dat 95% van de bestaande keyloggers in user-space draaien [42]. Ondanks de snelle groei van op keylogger gebaseerde fraude (dat wil zeggen, identiteitsdiefstal, wachtwoord lekkage, etc.), zijn er niet veel effectieve en efficiënte oplossingen voorgesteld om dit probleem aan te pakken. Traditionele detectiemechanismen gebruiken *fingerprinting* strategieën, vergelijkbaar met die welke worden gebruikt om virussen op te sporen. Helaas is deze strategie nauwelijks effectief tegen het grote aantal nieuwe keylogger-varianten dat elke dag in het wild opduikt.

In dit proefschrift pakken we het probleem van het opsporen van user-space keyloggers aan door juist hun specifieke gedrag uit te buiten. We maken vooral gebruik van het inzicht dat de activiteit van de keylogger strikt afhankelijk is van de input van de gebruiker. Door onze aanpak te generaliseren, verwerpen wij elke aanname over de manier waarop keyloggers zijn geprogrammeerd, en ontwikkelen we een black-box benadering die processen doorlicht door puur te kijken naar het vertoonde gedrag in het systeem. Om te voldoen aan het gemak van installatie en uitvoering van user-space keyloggers, hebben wij zowel oplossingen ontwikkeld die werken in situaties waar de gebruiker over weinig privileges beschikt, als oplossingen die werken in situaties waar de gebruiker geprivilegieerd is. De oplossingen die geen root privileges vereisen worden zelden in beschouwing genomen, omdat ze minder de neiging hebben efficiënt te zijn. Dit zijn de vele gevallen waarin de gebruiker een "niet-supergebruikersaccount" tot zijn beschikking heeft: Internetcafés, zakelijke laptops, of geleende terminals zijn hiervan allemaal goede voorbeelden. Wederom zijn wij hier ervan overtuigd dat een eerste lijn van verdediging verleend zal worden ongeacht de beschikbare privileges.

Dit proefschrift start met de discussie over KeySling en NoisyKey, twee oplossingen die user-space keyloggers kunnen opsporen en tolereren in situaties waarin de gebruiker beperkte privileges geniet. In KeySling zijn we op zoek naar samenhang tussen de I/O en gesimuleerde gebruikersactiviteiten. Als die correlatie bestaat, markeren we het als een keylogger. De reden hiervoor is, dat hoe intenser de stroom van toetsaanslagen is, hoe meer I/O-bewerkingen nodig zijn door de keylogger om de toetsaanslagen in een bestand op te slaan. NoisyKey verbetert de eerste lijn van verdediging van de gebruiker door hem of haar toe te staan samen te leven met een keylogger zonder zijn of haar privacy op het spel te zetten. Door het in ruis omhullen van de toetsaanslagen, gemaakt door de gebruiker, wordt de keylogger gevoed met willekeurige toetsaanslagen die niet uit elkaar kunnen worden gehouden van

de werkelijke toetsaanslagen van de gebruiker. Daarna introduceren we KLIMAX, een infrastructuur, draaiend in een omgeving waar de gebruiker geprivilegieerd is, dat in staat is om de geheugenactiviteit van een lopend proces te inspecteren. Deze nieuwe soort analyse maakt zelfs opsporing van keyloggers mogelijk die hun I/O verhullen. Dit is het geval bij privacy-schendende malware, oftewel spyware, dat zo minimaal mogelijk zijn activiteiten uitvoert om detectie te voorkomen. We besluiten deze dissertatie af door te kijken naar keyloggers die de vorm aannemen van applicatie *add-ons*, in plaats van de gebruikelijke losse applicatie. Gebaseerd op de nieuwe soort analyse die wordt aangeboden door KLIMAX, presenteren wij een nieuw "cross-browser" detectiemodel. Eenmaal geïnstalleerd, is het detectiemodel in staat is om *add-ons* te detecteren, waarmee we de webbrowser van de gebruiker omvormen tot een keylogger.