Threat Spotlight: GhostAdmin Malware

cylance.com/en_us/blog/threat-spotlight-ghostadmin.html

The BlackBerry Cylance Threat Research Team

Introduction

Data breaches are constantly in the news these days, targeting everyone from local restaurants to Fortune 500 companies. User accounts, names, credit card numbers, even your social security number are stored (you hope securely) in databases across the globe by numerous different companies. Your data is stored with your understanding that it will be protected and kept private from those who might abuse it. Unfortunately, time and time again, we discover that's not always the case.

With all these breaches in the media, it's easy to forget that big companies aren't the only target. Malware is just as capable of stealing a database on an enterprise server as it is stealing a single user's credentials and data from their personal laptop. You just don't typically hear about that kind of breach in the news. GhostAdmin 2.0 is a botnet found recently by MalwareHunterTeam (also known as Ghost iBot). It is capable of anything from stealing files to gaining full remote access, and all this is done using standard Windows libraries. Since its discovery in mid-January of 2017, new variants have been showing up on an almost weekly basis, and all with low conviction ratings by traditional antivirus (AV) products.

Code: Keylogging and Screen Capturing

Upon gaining entry, GhostAdmin doesn't waste any time, immediately starting a new thread for its keystroke logging (*Figure 1*). In order to store these logs, GhostAdmin attempts to hide them, pretending they are logs for the AV product Symantec Protection. It stores them in a folder created under "C:\Users\[current user]\AppData\Roaming\Symantec\".

Figure 1: GhostAdmin EntryPoint Getting Down to Business

The log is stored in an easily readable HTML file, going as far as recording keystrokes specific to the active window (*Figure 2*):

Figure 2: GhostAdmin's Usage of HTML for Keystroke Logs

Each time GhostAdmin runs, it downloads an update for its settings file. This settings file contains all the necessary command and control (C2) information for the client to connect to the newest server. This makes it easy for the server to be migrated regularly, making it easier to avoid detection-based AV on IP or URL blacklists. This settings file is base64-encoded and easily decodes into a standard format. The malware uses standard coding practices to read these settings from the file (Figure 3).

Here is an example of the settings information:

<quotebox>
FTP Server: accounts-security-settings(dot)com
FTP User: renard(at)accounts-security-settings.com
FTP Password: gho$t(at)dmin
IRC Server: irc.blafasel(dot)de
IRC Port: 6667
IRC Channel: #bobby7
Mail Sender: bucketshovel76(at)gmail.com
Mail Recipient: bucketshovel76(at)gmail.com
Mail Password: bigbobby12
FTP Folder: renard
Screen Image Timer: 300000000
Log File Timer: 36000000
GhostAdmin2.0
</quotebox>

Figure 3: Straightforward Processing of the Downloaded Settings File

GhostAdmin then sets a timer for one hour. This triggers an upload of the most recent logs to the FTP server. This transfer is straightforward, with a web client being created and used to upload the files (*Figure 4*). In this case, there is even an error handling step, throwing any errors into the C2 channel:

Figure 4: Simple WebClient Code to Upload Keystroke Logs

Creating yet another directory, GhostAdmin uses the 'CopyFromScreen' function from Drawing.dll to take complete captures of the user's screen. Each capture is indexed by user name and a timestamp and saved as a .jpg. Immediately after the screenshot is taken, the file is uploaded to the FTP server and deleted from the local host. These captures are taken on regular intervals as determined by the ScreenImageTimer variable in the settings.ini file once it is downloaded. Again, the author outputs any errors to the C2.

Code: IRC

The final thread created is for GhostAdmin's C2 channel and is used to establish an IRC channel. This portion of GhostAdmin accounts for, by far, the largest portion of the code and functionality. The exact IRC channel and server are derived from the settings.ini file downloaded by the second thread. Then a TcpClient instance connects and sends its logon messages to the C2 and even emails the author with Machine Name, User, IP, and client version (*Figure 5*).

Figure 5: Email Function to Notify Author That a GhostAdmin Bot is Online

After a quick 'PING' – 'PONG' to check for life on the IRC server, the client connects to the appropriate channel and starts listening for the bot master to send text starting with 'PRIVMSG,' indicating the master is giving a command. This IRC client is kept alive, reconnecting every one minute if it fails.

The list of commands is quite large in GhostAdmin. A full list is at the end of this blog, but for now, here is a high-level overview of the functions and their capabilities.

Host Information

GetIP, version, platform, checkfie, checkfolder, drives, tasklist, ipconfig, os, user, idletime.
These are functions designed to gather basic information about the host the client is running on, from the IP to currently running processes.

Data & Audio Exfiltration

Logfile, readfile, getfiles, uploadfile, screenshot, audio
These functions upload data to either the C2 or the FTP server.

The function AUDIO is worth noting, as it uses winmm.dll to access the machine's default microphone and start recording for the specified amount of time (*Figure 6*). This could be troublesome for an enterprise if the machine is kept in a location where sensitive information is discussed, for instance, near a corporate boardroom or in an executive team member's office. Typical of GhostAdmin, the audio file is immediately uploaded to the FTP server and then deleted from the local host.

Figure 6: AUDIO Command from GhostAdmin That Records From Default Microphone

Interaction

Turn(on/off)monitor, visit, download, delete, run, taskkill, kill, copy, enableremoteDesktop, shutdownWindows, restartWindows, (enable/disable)inputDevices, deleteLogs, deleteBrowserData, SQL, Update*
Functions capable of interacting with the infected host, these give the malicious actor several options for access. This includes browsing or downloading from URLs, killing the client or any process (taskkill), and restarting or powering off the host. EnableremoteDesktop modifies the registry value at 'HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server' to allow remote desktop connections to the infected host *(Figure 7*).

Figure 7: GhostAdmin Enables Remote Desktop via Registry Modification

DisableInputDevices
This uses the user32.dll function BlockInput to disable both mouse and keyboard interaction on the local host, essentially removing all control from the User. It is worth noting that in this situation hitting *Ctrl+Alt+Delete* will override this block. This is a security feature built into Windows, and throws a hard system error.

*SQL**
This group of functions allows the malicious actor to manipulate any SQL database that the infected user has access to. It first searches for any databases through System.data.sqlClient. If found it attempts login with credentials supplied through the IRC command. Once logged in, the ability to run select, insert, delete, and update are available.

DeleteBrowserData
This iterates through the 5 most common browsers, and deletes all session and user data (*Figure 8*). This is likely to force the user to log back into all their accounts, allowing GhostAdmin to capture as many of their credentials as possible.

Figure 8: GhostAdmin Can Delete Browser Data From Five Major Browsers

GhostAdmin: Incomplete

As it turns out, this appears to not be the final version of GhostAdmin, as there are a few functions that are marked as “coming soon” (*Figure 9*).

Figure 9: Apparently, GhostAdmin Even Has a Roadmap For Features!

Conclusion

GhostAdmin contains all the necessary functionality to gain complete control over a victim's computer. It is just as capable of capturing credentials from a single web browser form as it is modifying or downloading a large database.

With new variants being released with some frequency, and an incomplete codebase, GhostAdmin shows that it doesn't take complex malware to be effective. While we have been unable to find a live C2 to analyze, MalwareHunterTeam indicates that even with its small presence, hundreds of GBs of data have been stolen from both personal and corporate machines. We can assume that as this malware matures, variants will become both harder to detect and sent in even bigger deployments.

If you use our endpoint protection product, CylancePROTECT®, you were already protected from this attack. If you don't have CylancePROTECT, contact us to learn how our AI based solution can predict and prevent unknown and emerging threats.

List of IRC Commands

Complete list of commands client is capable of. This list is an exact copy of the array defined in the GhostAdmin code.

Figure 10: Command List Helpfully Provided in the Decompiled Version of GhostAdmin

Indicators of Compromise

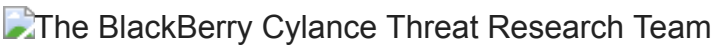
Configuration/Setting Files:
de60046e23435edf47ddc1cf1dd0fcb64b706e066d289b64855a38551ab3c4fe

Decoded:
FTP Server: secured-apps(dot)com
FTP User: ghostadmin(at)secured-apps.com
FTP Password: Z0di(at)c876
IRC Server: irc.synirc(dot)net
IRC Port = 6667
IRC Channel = #ghostadmin
Mail Sender: zodiacbot(at)gmail.com
Mail Recipient: zodiacbot(at)gmail.com
Mail Password: gho$t(at)dmin
f32c4a1ef662475840dd234249741b761be8d5a4eaaedc5c6d677f6e53d3a047

Decoded:
FTP Server: accounts-security-settings(dot)com
FTP User: renard(at)accounts-security-settings.com
FTP Password: gho$t(at)dmin
IRC Server: irc.blafasel(dot)de
IRC Port: 6667
IRC Channel: #bobby7
Mail Sender: bucketshovel76(at)gmail.com
Mail Recipient: bucketshovel76(at)gmail.com
Mail Password: bigbobby12
FTP Folder: renard
Screen Image Timer: 300000000
Log File Timer: 36000000
GhostAdmin2.0

SHA-256 Hashes:
a50d3218f4a6b7c89d3e8df3463ac3a4704d92acee57cfe8d79200ad0c887aa9
91374f78d11bdb0683f8145ef38645b4c1a5278d89fc07c5d8e94474c079b36f
dbe19b22364e17002fa43626fa04ef5d1b4938db84eae1c71c1f6c296b0ef560
5106d31eeb4e93e6c44d4637fee1a1e5c12c88ddacf668c58828184756bd78eb
834861a92c0fce4ee5bfe0a3b7000dff41f13ae4dab4d4d19a71686f531d98bd
af5725e808cc08c3b1b179fbc62c58c1d96ea6ceffa58cf2f498a32c56faa4a6
286701f6a7451d40b40bd05da0ef4d2f209e5e518b5a4ba4c119ab4bb788736c
3c9c2d1fe1b3fa6397c4440575086232cff61ed98b7cb82edf35773d4f253423

b46b2e67b86ff7fcd393984dfad5a23cf5442632abff26bc98f63f814678e428
bf0268523d3f0e50363eddde4ac593718e4ae893fc045afd2876253f5dc0b6e3
99c260ab07f31eff33c4b87d69c0bfc969f771bb8186dcbb4ac04473f42cf492
db5fc23fbd00449d8dafdee61760ea100651569e405b39c41eb0ec61acfb917b
fa6e3fcbb4d828a2de5375cdc706e6cf57f40f3aaa8c4a80d2ea19700bd63d37
36e0537e0d59016434085c8559c78e028a4022ebaf1c537c375c4687f66b47af
de60bfc9f1dcbaba111b87a707779fee4703ddfaa15ba4eb2e7df0520c403a46
bce2b4f0c8b9cefd5016a716ae2c47602855d68f686164b19b7c7f70d9ec8649
9af69d2d44640d568feaa3340acd84cd9baed19117e20d2f0a35f4f445086097
89f8cfb7e4db39d055c4264639325064a1d50c11ef8738e085a55f30a5f6b7bf



About The BlackBerry Cylance Threat Research Team

The BlackBerry Cylance Threat Research team examines malware and suspected malware to better identify its abilities, function and attack vectors. Threat Research is on the frontline of information security and often deeply examines malicious software, which puts us in a unique position to discuss never-seen-before threats.