कम्प्यूटर वायरस

कंप्यूटर प्रोग्राम जो खुद को दोहराने और फैलाने के लिए अन्य प्रोग्रामों को संशोधित करता है

कम्प्यूटर वायरस या **कम्प्यूटर विषाणु** एक कंप्यूटर प्रोग्राम *(computer program)* है जो अपनी अनुलिपि कर सकता है और उपयोगकर्ता की अनुमति के बिना एक कंप्यूटर को संक्रमित कर सकता है और उपयोगकर्ता को इसका पता भी नहीं चलता है। विभिन्न प्रकार के मैलवेयर *(malware)* और एडवेयर *(adware)* प्रोग्राम्स के सन्दर्भ में भी "वायरस" शब्द का उपयोग सामान्य रूप से होता है, हालाँकि यह कभी-कभी ग़लती से भी होता है। मूल वायरस अनुलिपियों में परिवर्तन कर सकता है, या अनुलिपियाँ ख़ुद अपने आप में परिवर्तन कर सकती हैं, जैसा कि एक रूपांतरित वायरस *(metamorphic virus)* में होता है। एक वायरस एक कंप्यूटर से दूसरे कंप्यूटर में तभी फ़ैल सकता है जब इसका होस्ट एक असंक्रमित कंप्यूटर में लाया जाता है, उदाहरण के लिए एक उपयोगकर्ता के द्वारा इसे एक नेटवर्क या इन्टरनेट पर भेजने से, या इसे हटाये जाने योग्य माध्यम जैसे फ्लॉपी डिस्क *(floppy disk)*, *CD (CD)*, या *USB* ड्राइव *(USB drive)* पर लाने से. इसी के साथ वायरस एक ऐसे संचिका तंत्र या जाल संचिका प्रमाली *(network file system)* पर संक्रमित संचिकाओं के द्वारा दूसरे कम्पूटरों पर फ़ैल सकता है जो दूसरे कम्प्यूटरों पर भी खुल सकती हों.कभी कभी कंप्यूटर का कीड़ा *(computer worm)* और ट्रोजन होर्सेस *(Trojan horses)* के लिए भी भ्रम पूर्वक वायरस शब्द का उपयोग किया जाता है। एक कीडा अन्य कम्प्यूटरों में ख़ुद फैला सकता है इसे पोषी के एक भाग्य के रूप में स्थानांतरित होने की जरुरत नहीं होती है और एक ट्रोजन होर्स एक ऐसी फाईल है जो हानिरहित प्रतीत होती है। कीडे और ट्रोजन होर्स एक कम्यूटर सिस्टम के आंकडों, कार्यात्मक प्रदर्शन, या कार्य निष्पादन के दौरान नेटवर्किंग को नुकसान पहुंचा सकते हैं। सामान्य तौर पर, एक कीड़ा वास्तव में सिस्टम के हार्डवेयर या सॉफ्टवेयर को नुकसान नहीं पहुंचाता, जबकि कम से कम सिद्धांत रूप में, एक ट्रोजन पेलोड, निष्पादन के दोरान किसी भी प्रकार का नुकसान पहुँचने में सक्षम होता है। जब प्रोग्राम नहीं चल रहा है तब कुछ भी नहीं दिखाई देता है लेकिन जैसे ही संक्रमित कोड चलता है, ट्रोजन होर्स प्रवेश कर जाता है। यही कारण है कि लोगों के लिए वायरस और अन्य मैलवेयर को खोजना बहुत ही कठिन होता है और इसीलिए उन्हें स्पायवेयर प्रोग्राम और पंजीकरण प्रक्रिया का उपयोग करना पड़ता है। आजकल अधिकांश व्यक्तिगत कंप्यूटर इंटरनेट और लोकर एरिया नेटवर्क से जुड़े हैं और लोकल एरिया नेटवर्क *(local area*

network), दूषित कोड को फैलाने की प्रक्रिया को सुविधाजनक बनाता है। आज का वायरस नेटवर्क सेवाओ का भी लाभ उठा सकता है जैसे वर्ल्ड वाइड वेब, ई मेल, त्वरित संदेश (*Instant Messaging*) और संचिका साझा (*file sharing*) प्रणालियां वायरसों और कीडों को फैलने में मदद करती हैं। इसके अलावा, कुछ स्रोत एक वैकल्पिक शब्दावली का उपयोग करते हैं, जिसमें एक वायरस स्व-अनुलिपि करने वाले मैलवेयर (*malware*) का एक रूप होता है। कुछ मेलवेयर, विनाशकारी प्रोग्रामों, संचिकाओं को डिलीट करने, या हार्ड डिस्क की पुनः फ़ॉर्मेटिंग करने के द्वारा कंप्यूटर को क्षति पहुचाने के लिए प्रोग्राम किए जाते हैं। अन्य मैलवेयर प्रोग्राम किसी क्षति के लिए नहीं बनाये जाते हैं, लेकिन साधारण रूप से अपने आप को अनुलिपित कर लेते हैं औए शायद कोई टेक्स्ट, वीडियो, या ऑडियो संदेश के द्वारा अपनी उपस्थिति को दर्शाते हैं। यहाँ तक की ये कम अशुभ मैलवेयर प्रोग्राम भी [उपयोगकर्ता (कम्प्यूटिंग)/कंप्यूटर उपयोगकर्ता] (*computer user*) के लिए समस्याएँ उत्पन्न कर सकते हैं।.वे आमतौर पर वैध कार्यक्रमों के द्वारा प्रयोग की जाने वाली कम्प्यूटर की स्मृति (*computer memory*) को अपने नियंत्रण में ले लेते हैं। इसके परिणामस्वरूप, वे अक्सर अनियमित व्यवहार का कारण होते हैं और सिस्टम को नुकसान पहुंचाते हैं। इसके अतिरिक्त, बहुत से मैलवेयर बग (*bug*) से ग्रस्त होते हैं, और ये बग सिस्टम को नुक्सान पंहुचा सकते हैं या डाटा क्षति (*data loss*) का कारण हो सकते हैं। कई सीआईडी प्रोग्राम ऐसे प्रोग्राम हैं जो उपयोगकर्ता द्वारा डाउनलोड किए गए हैं और हर बार पॉप अप किए जाते हैं। इसके परिणाम स्वरुप कंप्यूटर की गति बहुत कम हो जाती है लेकिन इसे ढूंढ़ना और समस्या को रोकना बहुत ही कठिन होता है।

मैकमैग वायरस 'यूनिवर्सल पीस', जैसा कि मार्च 1988 में एक मैक पर प्रदर्शित किया गया था

इतिहास

सबसे पहला वायरस क्रीपर था जो अरपानेट (*ARPANET*), पर खोजा गया, जो १९७० के दशक की शुरुआत में इंटरनेट से पहले आया था।[1] यह *TENEX* (*TENEX*) ऑपरेटिंग सिस्टम के द्वारा फैला और यह कंप्यूटर को नियंत्रित और संक्रमित करने के लिए किसी भी जुड़े मॉडम का उपयोग कर सकता था। यह संदेश प्रदर्शित कर सकता है कि "मैं क्रीपर हूँ; अगर पकड़ सकते हो तो मुझे पकडो". यह अफवाह थी कि रीपर प्रोग्राम, जो इसके कुछ ही समय बाद प्रकट होता है और क्रीपर की प्रतियाँ बनाता है और उन्हें हटा देता है, इसे संभवतया क्रीपर के निर्माता के द्वारा खेद पत्र में लिखा गया है। एक आम धारणा है कि एक प्रोग्राम जो "रोथर *J* (*Rother J*)"कहलाता था वह "इन दी वाइल्ड " प्रकट होने वाला पहला कंप्यूटर वायरस था –यह एक कंप्यूटर के बाहर या प्रयोगशाला में बनाया गया, लेकिन यह दावा ग़लत था। अन्य हाल ही के वायरसों के लिए जाने-माने कंप्यूटर वायरसों

और कीडों की समय रेखा *(Timeline of notable computer viruses and worms)* देखें.लेकिन यह " घर में " कम्प्यूटरों को संक्रमित करने वाला पहला वायरस था। १९८२ में रिचर्ड स्क्रेंता *(Richard Skrenta)*, के द्वारा लिखा गया, इसने ख़ुद को एप्पल *DOS (Apple DOS) 3.3* ऑपरेटिंग सिस्टम के साथ जोड़ लिया और फ्लॉपी डिस्क *(floppy disk)* के द्वारा फैला.[2]मूलतः यह वायरस एक हाई स्कूल के छात्र के द्वारा निर्मित एक मजाक था और इसे एक खेल के रूप में फ्लॉपी डिस्क पर डाल दिया गया। इसके ५० वें उपयोग पर *Elk Cloner (Elk Cloner)* वायरस सक्रिय हो गया, जिसने कंप्यूटर को संक्रमित किया और यह एक छोटी कविता को प्रर्दशित करता था " *Elk Cloner:The program with a personality*". इन दी वाइल्ड पहला पीसी वायरस एक बूट क्षेत्र का वायरस था जो *(c)* ब्रेन *((c)Brain)*[3] कहलाता था, इसे फारूक अल्वी ब्रदर्स *(Farooq Alvi Brothers)* के द्वारा १९८६ में बनाया गया, तथा लाहौर, पाकिस्तान के बाहर संचालित किया गया। इन्होने इस कथित वायरस को उनके द्वारा बनाये गए सोफ्टवेयर की प्रतियों कि चोरी रोकने के लिए बनाया.यद्यपि, विश्लेषकों ने दावा किया कि ***Ashar*** वायरस जो, ब्रेन की एक प्रजाति है, संभवतः वायरस के अन्दर कोड के आधार पर इसे पूर्व दिनांकित करता है। इससे पहले की कंप्यूटर नेटवर्क व्यापक होते अधिकांश वायरस हटाये जाने योग्य मध्यम *(removable media)*, विशेष रूप से फ्लॉपी डिस्क *(floppy disk)* पर फैल गए। शुरुआती दिनों में निजी कंप्यूटर *(personal computer)*, के कई उपयोगकर्ताओं के बीच नियमित रूप से जानकारी और प्रोग्रामों का विनिमय फ्लोपियों के द्वारा होता था। कई वायरस इन डिस्कों पर उपस्थित संक्रमित प्रोग्रामों से फैले, जबकि कुछ ने अपने आप को डिस्क के बूट क्षेत्र *(boot sector)* में इंस्टाल कर लिया, इससे यह सुनिश्चित हो गया की जब उपयोगकर्ता कंप्यूटर को डिस्क से बूट करेगा तो यह अनजाने में ही चल जाएगा.उस समय के पी सी पहले फ्लोपी से बूट करने का प्रयास करते थे, यदि कोई फ्लोपी ड्राइव में रह गई है। जब तक फ्लोपी का उपयोग कम नहीं हो गया तब तक यह सर्वाधिक सफल संक्रमण रणनीति थी, ***in the wild*** बूट क्षेत्र के वायरसों को बनाना सबसे आसान था।[4] पारंपरिक कंप्यूटर वायरस १९८० के दशक में उभरे, ऐसा निजी कंप्यूटर का उपयोग बढ़ने के कारण हुआ और इसके परिणाम स्वरुप *BBS (BBS)* और मॉडेम *(modem)* का उपयोग, तथा सॉफ्टवेयर का आदान-प्रदान बढ़ गया। बुलेटिन बोर्ड *(Bulletin board)* सॉफ्टवेयर के आदान प्रदान ने प्रत्यक्ष रूप से ट्रोजन होर्स प्रोग्राम्स को फैलाया और वायरस लोकप्रिय व्यावसायिक सॉफ्टवेयर को संक्रमित करने के लिए बनाये जाते थे।शेयरवेयर *(Shareware)* और बूटलेग *(bootleg)* ***BBS*** के वायरस के लिए आम वाहक *(vectors)* थे।होबिस्ट्स के "पाइरेट सीन" में जो खुदरा सॉफ्टवेयर *(retail software)*, की अवैध प्रतियों का व्य[पार कर रहे थे, व्यापारी आधुनिक अनुप्रयोगों और खेलों को जल्दी प्राप्त करना चाहते थे क्योंकि ये आसानी से वायरसों का लक्ष्य बनाये जा सकते थे। १९९० के दशक के मध्य के बाद से, मैक्रो वायरस *(macro virus)* आम हो गए .इनमें से अधिकांश वायरस माइक्रोसोफ्ट प्रोग्राम जैसे वर्ड *(Word)* औरएक्सेल *(Excel)* के लिए पटकथा भाषाओं में लिखे जाते हैं। ये वायरस दस्तावेज़ और स्प्रेडशीट को संक्रमित करते हुए माइक्रोसॉफ्ट ऑफिस में फ़ैल जाते हैं। चूंकि वर्ड और एक्सेल मैक *OS (Mac OS)* के लिए भी उपलब्ध थे, इसलिए इनमें से अधिकांश वायरस *Macintosh* कंप्यूटर *(Macintosh computers)* पर भी फ़ैल सकते थे। इनमें से अधिकांश वायरसों में संक्रमित ई मेल भेजने की क्षमता नहीं थी। जो वायरस ईमेल के माध्यम से फ़ैल सकते थे उन्होंने माइक्रोसॉफ्ट आउटलुक *(Microsoft Outlook)* *COM (COM)* इंटरफेस का फायदा उठाया. मैक्रो वायरस ने सॉफ्टवेयर का पता लगाने में अद्वितीय समस्या उत्पन्न कर दी.उदाहरण के लिए, माइक्रोसॉफ्ट वर्ड के कुछ संस्करणों ने मेक्रोस को अतिरिक्त रिक्त लाइनॉन के साथ अनुलिपित होने की इजाजत दे दी.वायरस समान रूप से व्यवहार करता था लेकिन इसे एक नए वायरस के रूप में पहचानने की भूल की जा रही थी। एक अन्य उदाहरण में, यदि दो मेक्रो वायरस एक दस्तावेज को एक साथ संक्रमित करते हैं और इन दोनों का संयोजन अपने आप को अनुलिपित कर सकता है, तो यह इन दोनों से "अलग" रूप में प्रकट होता है और एक वायरस होता है जो अपने " अभिभावकों " से अलग होता है।[5] एक वायरस एक संक्रमित मशीन पर सभी सम्पर्कों को त्वरित संदेश *(instant message)* के रूप में एक वेब पता *(web address)* लिंक भेज सकता है। यदि प्राप्तकर्ता, यह सोचता है कि लिंक एक मित्र

*(*एक विश्वसनीय स्रोत*)* से है, वह वेब साईट पर लिंक का अनुसरण करता है, साईट पर उपस्थित वायरस इस नए कंप्यूटर को संक्रमित करने में समर्थ हो सकता है और अपना प्रसार करने लगता है। वायरस परिवार की सबसे नई प्रजाति है क्रोस साईट पटकथा वायरस.यह वायरस अनुसंधान से उभरा और २००५ में इसका अकादमिक रूप से प्रदर्शन हुआ।[6]यह वायरस क्रोस साईट पटकथा *(**cross-site scripting**)* का उपयोग करके प्रसारित होता है। २००५ के बाद से ***in the wild***, क्रोस साईट पटकथा वायरसों के बहुत से उदहारण रहे हैं, सबसे उल्लेखनीय प्रभावित साईटें हैं माइस्पेस *(**MySpace**)* और याहू.

संक्रमण की रणनीतियां

अपने आप की अनुलिपि करने के लिए, एक वायरस को कोड का निष्पादन करने की तथा स्मृति पर लिखने की अनुमति होनी चाहिए इस कारण से, कई वायरस अपने आप को निष्पादन योग्य फाईलों से संलग्न कर लेते हैं जो उचित प्रोग्राम का हिस्सा हो सकता है। यदि एक उपयोग कर्ता एक संक्रमित प्रोग्राम को शुरू करने की कोशिश करता है तो, ऐसा हो सकता है की पहले वायरस का कोड निष्पादित हो. वाइरसों को दो प्रकार में विभाजित किया जा सकता है, उनके व्यवहार के आधार पर और उनके निष्पादन के आधार पर.अनिवासी वायरस तुंरत अन्य पोशियों को खोजते हैं जिन्हें संक्रमित किया जा सकता है, इन लक्ष्यों को संक्रमित करते हैं और अंततः नियंत्रण को अनुप्रयोग प्रोग्राम *(**application program**)* पर स्थानांतरित कर देते हैं जिसे उन्होंने संक्रमित किया है। निवासी वायरस पोशी की तलाश नहीं करते हैं जब वे शुरू होते हैं। इसके बजाय, एक निवासी वायरस अपने आप को निष्पादन पर स्मृति में लोड कर लेता है और नियंत्रण को पोषी प्रोग्राम पर स्थानांतरित कर देता है। वायरस पृष्ठ भूमि में सक्रिय रहता है और नए पोषियों को संक्रमित करता है जब इन फाईलों को अन्य प्रोग्रामों या ख़ुद ऑपरेटिंग सिस्टम के द्वारा एक्सेस किया जाता है।

अनिवासी वायरस

ऐसा माना जाता है कि अनिवासी वायरस एक *खोजक मॉड्यूल* और एक *प्रतिकृति मॉड्यूल* से युक्त होता है। खोजक मॉड्यूल संक्रमण हेतु नई फाईलें खोजने के लिए उत्तरदायी होता है। हर नई निष्पादन योग्य फाईल के लिए खोजक मॉड्यूल आक्रमण कर देता है, यह फाईल को संक्रमित करने के लिए अनुलिपि मॉड्यूल को बुलाता है।

निवासी वायरस

निवासी वायरस में एक अनुलिपि मॉड्यूल होता है जो एक अनिवासी वायरस के द्वारा नियोजित अनुलिपि मॉड्यूल के समान होता है। यद्यपि यह मोड्यूल खोजक मोड्यूल के द्वारा नहीं बुलाया जाता है। इसके बजाय, वायरस अनुलिपि मोड्यूल को स्मृति में लोड करता है जब इसका निष्पादन होता है और यह सुनिश्चित कर लेता है कि हर बार जब ऑपरेटिंग सिस्टम एक निश्चित आपरेशन का प्रदर्शन करता है तो इस मॉड्यूल का निष्पादन होता है। उदाहरण के लिए, हर बार जब ऑपरेटिंग सिस्टम एक संचिका को निष्पादित करता है, अनुलिपि मोड्यूल को बुलाया जा सकता है। इस मामले में, वायरस कंप्यूटर पर निष्पादित होने वाले प्रत्येक उपयुक्त प्रोग्राम को संक्रमित कर देता है। कभी कभी निवासी वायरस को आगे दो श्रेणियों में उपविभाजित किया जाता है एक *तीव्र संक्रामक* और दूसरे *धीमें संक्रामक.*तीव्र संक्रामक को इस प्रकार से बनाया गया है कि जितनी अधिक संचिकाओं को हो सके संक्रमित कर सके. उदाहरण के लिए, एक तीव्र संक्रामक अक्सेस हो सकने वाली हर सम्भव पोषी

संचिका को संक्रमित करता है। यह एक वायरस रोधी सॉफ्ट वेयर के लिए एक विशेष समस्या है, चूँकि एक वायरस स्केनर हर सम्भव पोषी संचिका को अक्सेस कर लेगा जब यह सिस्टम का व्यापक स्केन करता है। यदि वायरस स्केनर इस बात का ध्यान नहीं रख पाता कि ऐसा एक वायरस स्मृति में उपस्थित है, वायरस स्केनर पर पीछे से वार करता है और इस प्रकार से स्केन हो रही सभी संचिकाओं को संक्रमित कर देता है। तीव्र संक्रामक अपने प्रसार के लिए तीव्र संक्रमण दर पर भरोसा करते हैं। इस विधि का नुकसान यह है कि कई संचिकाओं को संक्रमित करने से पता लगाने की संभावना अधिक हो जाती है, क्यों कि वायरस कम्पूटर को धीमा कर देता है या कई संदिग्ध क्रियाओं को करता है जो वायरस विरोधी सॉफ़्टवेयर के द्वारा देखी जा सकती हैं। दूसरी और धीमे संक्रामक, इस प्रकार से बनाये जाते हैं कि वे पोषी को अन-आवृत रूप से संक्रमित करते हैं। उदाहरण के लिए, कुछ धीमे संक्रामक संचिकाओं को केवल तभी संक्रमित करते हैं जब उनको कॉपी किया जा रहा है। धीमे संक्रामक कि सीमित क्रिया कि वजह से उनका पता लगाना आसान नहीं होता है: वे कंप्यूटर को बहुत अधिक धीमा नहीं करते हैं, पर बार-बार वायरस विरोधी सॉफ्ट वेयर को झटका देते हैं जिसे प्रोग्रामों के संदिग्ध व्यवहार के द्वारा पहचाना जा सकता है। धीमे संक्रामक बहुत अधिक सफल प्रतीत नहीं होते हैं,

वाहक और पोषी

वायरसों ने भिन्न प्रकार के प्रसार माध्यमों या पोषियों को लक्ष्य बनाया है। यह सूची ख़त्म नहीं हो सकती है:

- बाइनरी निष्पादन योग्य संचिका *(executable file)* (जैसे कि *COM* संचिका *(COM file)* और *EXE (EXE) MS-*डॉस *(MS-DOS)* में संचिकाएँ, पोर्टेबल निष्पादन योग्य *(Portable Executable)* संचिकाएँ माइक्रोसॉफ्ट विन्डोज़ *(Microsoft Windows)* में, और *ELF (ELF)* संचिकाएँ *Linux* में)
- फ्लॉपी डिस्क *(floppy disk)* का आयतन बूट रिकॉर्ड *(Volume Boot Record)* और हार्ड डिस्क विभाजन
- एक हार्ड डिस्क का मास्टर बूट रिकॉर्ड *(master boot record) (MBR)*
- सामान्य उद्देश्य की पटकथा *(script)* संचिकाएँ (जैसे बैच संचिका *(batch file) MS* - डॉस *(MS-DOS)* में और माइक्रोसॉफ्ट विन्डोज़ *(Microsoft Windows), VBScript (VBScript)* संचिकाएँ, और शेल *script (shell script)* संचिकाएँ यूनिक्स की तरह *(Unix-like)* के प्लेटफार्म पर) .
- अनुप्रयोग विशिष्ट स्क्रिप्ट संचिकाएँ (जैसे *Telix (Telix)*- लिपियाँ)
- दस्तावेज जिनमें मेक्रोस *(macro)* हो सकते हैं (जैसे माइक्रोसॉफ्ट वर्ड *(Microsoft Word)* दस्तावेज, माइक्रोसॉफ्ट एक्सेल *(Microsoft Excel)* स्प्रेडशीट, *AmiPro (AmiPro)* दस्तावेज, और माइक्रोसॉफ्ट एक्सेस *(Microsoft Access)* डेटाबेस संचिकाएँ)
- क्रोस साईट पटकथा *(Cross-site scripting)* वेब अनुप्रयोगों में कमजोरियां
- मनमानी कंप्यूटर संचिकाएँ. एक दोहन योग्य बफर अतिप्रवाह *(buffer overflow)*, प्रारूप स्ट्रिंग *(format string)*, रेस की स्थिति *(race condition)* या अन्य प्रोग्राम में दोहन योग्य एक बग जो संचिका को पढता है इसके भीतर छुपे हुए कोड के निष्पादन पर हमला कर सकता है। इस प्रकार के अधिकांश बग कंप्यूटर वास्तुकला *(computer architecture)* में

दोहन के लिए अधिक जटिल बनाये जा सकते हैं इसमें सुरक्षा लक्षण जैसे एक निष्पादित निष्क्रिय बिट *(execute disable bit)* और / या लेआउट *randomization* होते हैं।

PDFs, जैसे *HTML*, दुर्भावनापूर्ण कोड से *सम्बंधित* हो सकता है। कुछ वायरस लेखकों ने जो लिखा है उसका कोई महत्त्व नहीं है।*EXE* विस्तार के अंत पर*PNG* (उदाहरण के लिए), उम्मीद है कि उपयोगकर्ता विश्वस्त फाईल के प्रकार पर रुक जाएगा, वह इस बात पर ध्यान नहीं देगा की कंप्यूटर अंतिम प्रकार की संचिका के साथ *प्रारंभ* होगा.(कई ऑपरेटिंग सिस्टम ज्ञात संचिका प्रकार के विस्तार को बाई डिफॉल्ट छुपा लेते हैं, इसलिए उदाहरण के लिए एक संचिका का नाम जो "*.png.exe*" पर ख़त्म होता है,"*.png*" पर ख़त्म होता हुआ प्रर्दशित किया जाएगा.) ट्रोजन हार्स (कम्प्यूटिंग) *(Trojan horse (computing))* देखें

पता लगाने से बचने के लिए विधियां

उपयोगकर्ता के द्वारा पता लगाने से बचने के लिए, कुछ वाईरस विभिन्न प्रकार के छल करते हैं। कुछ पुराने वायरस, विशेष रूप से *MS - DOS* मंच पर, यह सुनिश्चित कर लेते हैं कि जब संचिका को वायरस के द्वारा संक्रमित किया जाता है तो पोषी संचिका के " अंतिम बार संशोधित " होने की दिनांक बनी रहती है। यद्यपि यह दृष्टिकोण वायरस विरोधी सॉफ़्टवेयर को मुर्ख नहीं बनता है, विशेष रूप से वो जो संचिका परिवर्तन पर चक्रीय अतिरेक की जाँच *(Cyclic redundancy check)* कि दिनांकों को बनाये रखता है। कुछ वायरस अपने आकार को बढाये बिना और संचिकाओं को क्षति पहुचाये बिना संचिकाओं को संक्रमित कर सकते हैं। वे ऐसा निष्पादन योग्य संचिकाओं के अप्रयुक्त क्षेत्रों में अधिलेखन के द्वारा करते हैं। ये *केविटी वायरस* कहलाता है। उदाहरण के लिए की *CIH* वायरस *(CIH virus)*, या चेरनोबिल वायरस *(Chernobyl Virus)*, पोर्टेबल निष्पादन योग्य *(Portable Executable)* संचिकाओं को संक्रमित करता है। क्योंकि उन संचिकाओं में कई खाली स्थान थे, वायरस जो १ *KB (KB)* लंबाई का था, संचिका के आकर में नहीं जुड़ा. कुछ वायरस वायरस विरोधी सॉफ्टवेयर के कार्य को ख़त्म करके अपने आप को प्रकट नहीं होने देते हैं, इससे पहले कि वह उसका पता लगा ले. क्योंकि कंप्यूटर और ऑपरेटिंग सिस्टम अधिक विकसित और जटिल हो रहे हैं, छुपाने कि पुरानी तकनीकों को नवीनीकृत करने या प्रतिस्थापित करने कि आवशयकता है। एक कंप्यूटर को वायरस से सुरक्षित रखने कि मांग यह हो सकती है कि संचिका तंत्र हर प्रकार के संचिका एक्सेस के लिए विस्तृत और स्पष्ट अनुमति की और पलायन कर जाए.

बैट संचिकाओं और अन्य अवांछनीय पोषियों को टालना

एक वायरस के लिए आगे फैलने के लिए पोषियों को संक्रमित करने कि जरुरत होती है। कुछ मामलों में, एक पोषी को संक्रमित करना एक खराब विचार हो सकता है उदाहरण के लिए, कई वायरस विरोधी प्रोग्राम अपने कोड की अखंडता की जांच करते हैं। इसलिए ऐसे प्रोग्रामों का संक्रमण इस सम्भावना को बढ़ाएगा कि इस वायरस का पता चल गया है। इस कारण से, कुछ वायरस उन प्रोग्रामों को संक्रमित नहीं करते हैं जो वायरस विरोधी सॉफ़्टवेयर का हिस्सा है। एक अन्य प्रकार का पोषी जिससे वायरस कभी कभी बचने कि कोशिश करता है वह है *बैट संचिकाएँ*.बैट संचिकाएँ *(या गोट संचिकाएँ)* वे संचिकाएँ हैं जो कि विशेष रूप से वायरस विरोधी सॉफ़्टवेयर या ख़ुद वायरस विरोधी पेशेवरों के द्वारा एक वायरस के द्वारा संक्रमित होने के लिए बनाई गई हैं। इन संचिकाओं को भिन्न कारणों से बनाया जा सकता है, ये सभी वायरस का पता लगाने से सम्बंधित हैं।

- वायरस विरोधी पेशेवर बैट संचिकाओं का उपयोग एक वायरस के नमूने लेने के लिए कर सकते हैं। *(*अर्थात एक ऐसे प्रोग्राम संचिका कि कॉपी जो वायरस के द्वारा संक्रमित है।*)* एक छोटी संक्रमित बैट संचिका का संग्रहण और विनिमय अधिक प्रायोगिक है, बजाय एक बड़े अनुप्रयोग प्रोग्राम के जो वायरस के द्वारा संक्रमित है।
- वायरस विरोधी पेशेवर बैट संचिकाओं का उपयोग वायरस के व्यवहार का अध्ययन करने के लिए और जांच विधियों के मूल्यांकन के लिए करते हैं। यह विशेष रूप से उपयोगी है जब वायरस बहुरूपी *(polymorphic)* हो. इस मामले में, वायरस बड़ी संख्या में बैट संचिकाओं को संक्रमित कर सकता है। संक्रमित संचिकाओं को उपयोग यह पता लगाने के लिए किया जाता है कि एक वायरस स्केनर वायरस के सभी संस्करणों का पता लगता है या नहीं.
- कुछ वायरस विरोधी सॉफ़्टवेयर ऐसी बैट संचिकाओं पर जाते हैं जो नियमित रूप से एक्सेस होती हैं। जब ये संचिकाएँ संशोधित की जाती हैं, वायरस विरोधी सॉफ़्टवेयर चेतावनी देता है कि शायद एक वायरस सिस्टम पर सक्रिय है।

चूँकि बैट संचिकाओं का उपयोग वायरस का पता लगाने में या जांच को सम्भव बनने में किया जाता है, वायरस के द्वारा संक्रमण नहीं होने का लाभ उठाया जा सकता है। आम तौर पर वाइरस आम तौर पर ऐसा वाइरस प्रोग्रामों से बच कर ऐसा करते हैं, जैसे छोटी प्रोग्राम संचिकाएँ या प्रोग्राम जिनमें ' कूड़ा निर्देश ' के विशिष्ट प्रतिरूप हों. बैटिंग को मुश्किल बनाने के लिए एक संबंधित रणनीति है *विरल संक्रमण.*कभी कभी, विरल संक्रामक एक पोषी संचिका को संक्रमित नहीं करते हैं जो अन्य स्थितियों में संक्रमण के लिए उपयुक्त उम्मीदवार हो सकता है। उदाहरण के लिए, एक वायरस यादृच्छिक आधार पर तय कर सकता है कि एक संचिका को संक्रमित करना है या नहीं, या एक वायरस सप्ताह के विशेष दिनों में ही पोषी संचिकाओं को संक्रमित कर सकता है।

चोरी

कुछ वायरस ऑपरेटिंग सिस्टम से अनुरोध करके वायरस विरोधी सॉफ़्टवेयर को धोखा देने कि कोशिश करते हैं। वायरस विरोधी सॉफ़्टवेयर संचिका को पढने का अनुरोध करता है और यह अनुरोध *OS* के बजाय वायरस को जाता है, इस प्रकार से एक वायरस अपने आप को छुपा लेता है। अब वायरस विरोधी सॉफ़्टवेयर की संचिका के असंक्रमित संस्करण पर लोट जाता है, जिससे कि ऐसा प्रतीत होता है कि संचिका " स्वच्छ " है। आधुनिक वायरस विरोधी सॉफ़्टवेयर के पास वायरस की चोरी का पता लगाने के लिए कई तकनीकें हैं। चोरी को रोकने की केवल एक विश्वसनीय विधि है एक ऐसे माध्यम से बूट करना जो स्वच्छ हो.

स्वयं संशोधन

अधिकांश वायरस विरोधी प्रोग्राम साधारण प्रोग्रामों के भीतर वायरस प्रतिरूप खोजने कि कोशिश करते हैं इसे तथाकथित *वायरस के हस्ताक्षर* कहा जाता है। हस्ताक्षर एक लाक्षणिक बाईट प्रतिरूप है, जो एक विशेष वायरस या वायरस परिवार का एक भाग है। यदि एक वायरस स्केनर एक संचिका में ऐसा प्रतिरूप खोज लेता है, यह उपयोगकर्ता को सूचित कर देता है कि संचिका संक्रमित है। उपयोगकर्ता फिर इस संचिका को नष्ट कर सकते हैं, या *(*कुछ मामलों में*)* संक्रमित संचिका को " शुद्ध " या " ठीक "कर सकते हैं। कुछ वायरस ऐसी तकनीकों का उपयोग करते हैं कि जो हस्ताक्षर के द्वारा इसका पता लगाने को मुश्किल ही नहीं बल्कि असंभव बना देता है। ये वायरस प्रत्येक संक्रमण पर अपना कोड बदल लेते हैं। अर्थात प्रत्येक संक्रमित संचिका में वायरस की एक अलग प्रजाति होती है।

एक चार कुंजी के साथ एन्क्रिप्शन.

वायरस के कूटलेखन के लिए एक और अधिक उन्नत विधि है सरल एन्क्रिप्शन *(encryption)* का उपयोग.इस मामले में, वायरस एक छोटे से *decrypting* मॉड्यूल और वायरस के कोड कि एक एन्क्रिप्टेड कॉपी से युक्त होता है। यदि प्रत्येक संक्रमित संचिका के लिए वायरस एक अलग कुंजी से एन्क्रिप्टेड है, तो वायरस का केवल एक भाग स्थिर बना रहता है वह है *decrypting* मॉड्यूल, जो *(*उदाहरण के लिए*)* के अंत से संलग्न होगा.इस मामले में, एक वायरस स्कैनर हस्ताक्षर का उपयोग करके सीधे वायरस का पता नहीं लगा सकता, लेकिन यह फ़िर भी डीक्रिप्टिंग मॉड्यूल का पता लगा सकता है, जो संभव वायरस की अभी भी अप्रत्यक्ष जांच कर सकता है। चूंकि ये संक्रमित पोषी पर संग्रहित कुंजिया होंगी, वास्तव में अन्तिम वायरस को विकोड करना पूरी तरह से सम्भव है, लेकिन इसकी शायद जरुरत नहीं है, चूँकि स्वयं संशोधित कोड इतना दुर्लभ है कि इसके कारण से वायरस स्केनर कम से कम संदिग्ध संचिका को अंकित कर सकता है। एक पुराने लेकिन कम्पैक्ट, एन्क्रिप्शन में एक स्थिर वायरस कि प्रत्येक बाईट *XORing (XORing)* शामिल है, ताकि ऑपरेशन का दोहरान केवल डिक्रिप्शन के लिए हो.यह संदिग्ध कोड है जो अपने आप को संशोधित करता है, इसलिए कई वायरस परिभाषाओं में एन्क्रिप्शन / डिक्रिप्शन हस्ताक्षर का एक भाग हो सकता है।

बहुरूपी कोड

बहुरूपी कोड *(Polymorphic code)* पहली तकनीक थी जो वायरस स्केनर के लिए एक गंभीर खतरा बनी. नियमित एन्क्रिप्टेड वायरस की तरह, एक बहुरूपी वायरस अपने एन्क्रिप्टेड कोड के साथ संचिकाओं को संक्रमित करता है, जो एक डिक्रिप्शन मॉड्यूल के द्वारा डिकोड कर दिया जाता है। बहुरूपी वायरस के मामले में, यह डिक्रिप्शन मॉड्यूल भी प्रत्येक संक्रमण पर संशोधित होता है। इसलिए ठीक प्रकार से लिखे गए बहुरूपी वायरस में ऐसे कोई भाग नहीं हैं जो संक्रमणों में समान हों, यह हस्ताक्षर का उपयोग करके इसका पता लगाना मुश्किल बनाता है। वायरस विरोधी सॉफ़्टवेयर एक प्रतिद्वंद्वी का उपयोग करके *decrypting* के द्वारा इसका पता लगा सकता है, या एन्क्रिप्टेड वायरस शरीर के सांख्यिकीय प्रतिरूप विश्लेषण के द्वारा इसका पता लगा सकता है। बहुरूपी कोड को सक्षम करने के लिए, वायरस के पास इसके एन्क्रिप्टेड शरीर में कहीं पर एक बहुरूपी इंजन *(*इसे उत्परिवर्तन इंजन भी कहा जाता है*)* होना चाहिए .इस तरह के इंजन कैसे काम करते है इस पर तकनीकी विस्तार के लिए बहुरूपी कोड *(Polymorphic code)* देखें. कुछ वायरस बहुरूपी कोड को इस प्रकार से काम में लेते हैं कि यह मुख्य रूप से वायरस के उत्परिवर्तन की दर को बाध्य करता है। उदाहरण के लिए, एक वायरस केवल कुछ ही समय में उत्परिवर्तित होने के लिए प्रोग्राम किया जाता है, या इसे इस प्रकार से प्रोग्राम किया जा सकता है कि जब ये कंप्यूटर पर ऐसी संचिका को संक्रमित कर रहा हो जिसमें पहले से वायरस की अनुलिपियाँ हो, तो यह उत्परिवर्तन से बच सके. जिसे धीमे बहुरूपी कोड का उपयोग करने का लाभ यह है कि इससे वायरस विरोधी पेशेवर के लिए वायरस का प्रतिनिधि का नमूना प्राप्त करना मुश्किल हो जाता है, क्योंकि बैट संचिकाएँ जो एक बार चलने में संक्रमित हो जाती हैं, उनमें प्रारूपिक रूप से वायरस का समान नमूना होगा. इसे यह सम्भावना बढ़ जायेगी कि वायरस स्केनर के द्वारा कि गई जांच अविश्वसनीय होगी और यह हो सकता है कि वायरस के कुछ उदाहरण जांच का पता लगाने में सक्षम न हों.

रूपांतरित कोड

अनुकरण के द्वारा जांच से बचने के लिए कुछ वायरस हर बार जब एक नए निष्पादनयोग्य को संक्रमित करते हैं तब अपने आप को पुनर्लिखित कर लेते हैं। वायरस जो ऐसी तकनीक का उपयोग करते हैं वे रूपांतरित *(metamorphic)* कहलाते हैं।

रूपांतरण *(metamorphism)* को संभव बनाने के लिए एक **रूपांतरित इंजन** की जरूरत होती है। एक रूपांतरित वायरस आमतौर पर बहुत बड़ा और जटिल होता है। उदाहरण के लिए, *W32/Simile (W32/Simile)* में समूह भाषा कोड की १४००० से अधिक रेख्यें होती हैं, जिनमें से ९०% रूपांतरित इंजन का हिस्सा है।[7]

भेद्यता और मापने के लक्षण

वायरस के लिए ऑपरेटिंग सिस्टम की भेद्यता

जैसे एक आबादी में आनुवंशिक विविधता *(genetic diversity)* के कम होने पर उसमें एक मात्र रोग की सम्भावना कम हो जाती है, समान रूप से एक नेटवर्क पर सॉफ्टवेयर सिस्टम कि विविधता की भी वायरस की विनाशकारी क्षमता को सीमित करती है। यह १९९० में विशेष विचार का विषय बन गया जब माइक्रोसॉफ्ट ने डेस्कटॉप ऑपरेटिंग सिस्टम और कार्यालय सुइट *(office suite)* में बाजार में प्रभुत्व प्राप्त कर लिया। माइक्रोसॉफ्ट सॉफ़्टवेयर के उपयोगकर्ता *(*विशेषकर नेटवर्किंग सॉफ्टवेयर जैसे माइक्रोसॉफ्ट आउटलुक *(Microsoft Outlook)* और इंटरनेट एक्सप्लोरर *(Internet Explorer))* विशेष रूप से वायरस के प्रसार के लिए कमजोर हैं। माइक्रोसॉफ्ट सॉफ़्टवेयर को वायरस के द्वारा लक्ष्य बनाये जाने का कारण है उनका डेस्कटॉप प्रभुत्व होना और अक्सर कई गलतियों और वायरस लेखकों के लिए छिद्रों कि वजह से इसकी आलोचना की जाती है। समन्वित और गैर एकीकृत माइक्रोसॉफ्ट अनुप्रयोग *(*जैसे माइक्रोसॉफ्ट ऑफिस*)* और संचिका प्रणाली के एक्सेस के साथ भाषाओँ कि पटकथाओं के अनुप्रयोग *(*उदाहरण के लिए दृश्य मूल लिपि *(Visual Basic Script) (VBS)*, और नेटवर्किंग लक्षणों के साथ अनुप्रयोग*)* भी विशेष रूप से जोखिम युक्त हैं। यद्यपि विंडोस वायरस के लेखकों के लिए, सबसे लोकप्रिय ऑपरेटिंग सिस्टम है, कुछ वायरस दूसरे प्लेटफार्म पर भी मौजूद है। कोई भी ऑपरेटिंग सिस्टम जो चलने के लिए तीसरे पक्ष के प्रोग्राम को अनुमति दे सकता है वह सैद्धांतिक रूप से वायरस को चला सकता है। कुछ ऑपरेटिंग सिस्टम दूसरों के मुकाबले कम सुरक्षित हैं .यूनिक्स आधारित *OS (*और *NTFS* को जानने वाले अनुप्रयोग विन्डोज़ *NT* आधारित प्लेटफार्म पर*)* केवल अपने उपयोग कर्ताओं को ही अपने निर्देशों के अंतर्गत सुरक्षित स्थान के भीतर निष्पादन की अनुमति देते हैं। इंटरनेट आधारित एक शोध से पता चला कि ऐसे मामले भी थे जब लोगों ने जान-बूझ कर एक वायरस को डाऊनलोड करने के लिए एक विशेष बटन को दबाया.एक सुरक्षा फर्म *F-Secure* ने एक ६ माह का अभियान गूगल ऐडवर्ड्स *(Google AdWords)* पर चलाया जिसने कहा "क्या आपका पी सी वायरस मुक्त है?यह यहाँ संक्रमित प्राप्त करें *!* " .इसका परिणाम था ४०९ क्लिक.[8]२००६ में अपेक्षाकृत कुछ सुरक्षा के तरीके थे[9]जो मैक *OS X (Mac OS X)* को लक्ष्य बना रहे थे *(*एक यूनिक्स आधारित संचिका प्रणाली और कर्नेल *(kernel)* के साथ*)* . पुराने एप्पल ऑपरेटिंग सिस्टम के लिए वायरस कि एक संख्या जो मैक ओएस क्लासिक के नाम से जानी जाती है, वो अलग अलग स्रोतों में बहुत अलग होती है, एप्पल के साथ कहा जाता है कि केवल चार प्रकार के जाने माने वायरस होते हैं और स्वतंत्र सूत्रों *(independent sources)* के अनुसार ६३ वायरस के रूप हैं। यह कहना सुरक्षित है कि बाजार के कम शेयर कि वजह से *Macs* को लक्ष्य बनाने की सम्भावना कम होती है और इस प्रकार से एक मैक विशेष वायरस केवल कम्प्यूटरों के एक छोटे अनुपात को ही संक्रमित कर सकता है। *(*प्रयास को कम वांछनीय बनते हुए*)* .मेक्स और विंडोस के बीच वायरस का जोखिम मुख्य बिक्री बिन्दु है, एक यह कि *Apple (Apple)* उनके एक मैक प्राप्त करें *(Get a Mac)* विज्ञापन[10] में काम लेता है। इस के अनुसार मेक्स में भी माइक्रोसॉफ्ट विन्डोज़ कि तरह सुरक्षा के मुद्दे होते हैं, यद्यपि किसी ने भी *in the wild* सफलता पूर्वक इसका पूर्ण लाभ नहीं उठाया है। विंडोस और यूनिक्स में समान पटकथा की क्षमता है, लेकिन जब यूनिक्स स्वाभाविक रूप से सामान्य उपयोगकर्ताओं को ऑपरेटिंग सिस्टम वातावरण के लिए परिवर्तन करने के लिए एक्सेस करने से रोकता है, विन्डोज़ की पुरानि प्रतियां जैसे विन्डोज़ ९५ और ९८ ऐसा

नहीं करती.१९९७ में, जब एक "*Bliss (Bliss)*" नामक वायरस *Linux* के लिए जारी किया गया था --प्रमुख वायरस विरोधी विक्रेताओं ने एक चेतावनी जारी की यूनिक्स की तरह *(Unix-like)* के सिस्टम ठीक विडोस की तरह वायरस का शिकार बन सकते हैं।[11]ब्लिस वायरस को वायरसों का लाक्षणिक माना जा सकता है--यूनिक्स प्रणालियों पर --कीडों के विपरीत.ब्लिस की जरुरत है कि उपयोगकर्ता इसे स्पष्ट रूप से चलाता है *(इसलिए यह एक ट्रोजन है)* और यह केवल ऐसे प्रोग्रामों को संक्रमित कर सकता है जो उपयोगकर्ता के द्वारा संशोधित किए जा सकते हैं। विन्डोज़ उपयोगकर्ता के विपरीत, अधिकांश यूनिक्स उपयोगकर्ता किसी सॉफ्टवेयर को इन्सटौल करने के अलावा किसी व्यवस्थापक उपयोगकर्ता के रूप में लॉग इन *(log in)* नहीं करते: यहाँ तक कि यदि एक उपयोगकर्ता वायरस को चलाता है तो भी यह ऑपरेटिंग सिस्टम को नुकसान नहीं कर सकता है। ब्लिस वायरस कभी भी व्यापक नहीं हुआ और मुख्य रूप से अनुसंधान के लिए उत्सुकता का विषय बना रहा.इसके निर्माता ने बाद में यूज़नेट को इसका स्रोत कोड पोस्ट किया, शोधकर्ताओं को यह जानने कि अनुमति दी कि यह कैसे कम करता है।[12]

सॉफ्टवेयर विकास की भूमिका

चूँकि सिस्टम स्रोत का अनाधिकृत उपयोग रोकने के लिए सॉफ्टवेयर को अक्सर सुरक्षा व्यवस्था के साथ डिजाईन किया जाता है, कई वायरस एक सिस्टम या अनुप्रयोग में सॉफ्टवेयर बग *(software bug)* को फैलने में मदद करते हैं।सॉफ्टवेयर विकास *(Software development)* रणनीतियां जो बड़ी संख्या में बग का उत्पादन करती हैं, सामान्य रूप से सक्षम दोहन भी उत्पन्न करती है।

वायरस विरोधी सॉफ्टवेयर और अन्य रोकथाम के उपाय.

कई उपयोग कर्ता वायरस विरोधी सॉफ्टवेयर *(*ऐंटिवायरस सॉफ़्टवेयर, *anti-virus software)* इंस्टाल करते हैं, जो कंप्यूटर डाउनलोड *(download)* के बाद या निष्पादन योग्य के चलने के बाद ज्ञात वायरस का पता लगा लेते हैं या उसे नष्ट कर देते हैं। ऐसे दो सामान्य तरीके हैं जिन्हें एक वायरस विरोधी सॉफ्टवेयर *(anti-virus software)* अनुप्रयोग वायरस की जांच करने के लिए काम में लेता है। वायरस की जांच की पहली और सबसे सामान्य विधि है वायरस के हस्ताक्षर *(virus signature)* परिभाषा की सूची का उपयोग.यह कंप्यूटर की स्मृति के अवयवों *(*अपने *RAM (RAM)* और बूट क्षेत्र *(boot sector))* और स्थायी या अस्थायी ड्राइवों *(*हार्ड ड्राइव, फ्लॉपी ड्राइव*)* पर संगृहीत संचिकाओं की जांच के द्वारा, कार्य करता है तथा इन संचिकाओं की तुलना ज्ञात वायरस के "हस्ताक्षर" के डेटाबेस के ख़िलाफ़ की जाती है। जांच की इस विधि का नुक्सान यह है कि उपयोग कर्ता केवल वायरस से सुरक्षित रहता है जो अपनी पिछली परिभाषा का अद्यतन करता रहता है। दूसरी विधि है अनुमानी *(heuristic)* कलन विधि जिसमें सामान्य व्यवहार पर आधारित वायरसों का पता लगाया जाता है। इस विधि में वायरसों का पता लगाने कि क्षमता होती है जिसके लिए अभी भी वायरस विरोधी कंपनियों को हस्ताक्षर बनाना है। कुछ वायरस विरोधी प्रोग्राम खुली हुई संचिकाओं को स्केन करने में सक्षम होते हैं साथ ही समान तरीके से '*on the fly*' भेजे गए और प्राप्त किए गए ई-मेल्स को भी स्केन कर सकते हैं। इसे "*on-access scanning*"कहते हैं। वायरस विरोधी सोफ्टवेयर पोशी सॉफ्टवेयर कि वायरस को प्रेषित करने की क्षमता को परिवर्तित नहीं करता है। उपयोगकर्ता को पैच *(patch)* सुरक्षा छिद्र के लिए अपने सोफ्टवेयर नियमित रूप से अद्यतन करने चाहिए.एंटी वायरस सॉफ्टवेयर को भी नियमित रूप से अद्यतन किए जाने की आवश्यकता है ताकि आधुनिक खतरों को रोका जा सके. कोई भी व्यक्ति भिन्न मिडिया पर डाटा *(*और आपरेटिंग सिस्टम*)* के नियमित रूप से बैकअप *(backup)* से वायरस के द्वारा की गई क्षति को कम कर सकता है, इन्हें या तो सिस्टम से असंबद्ध *(*ज्यादातर समय*)*, रखा जाता है, या अन्य कारणों जैसे भिन्न संचिका प्रणालियों *(file system)* का उपयोग के लिए

इसे नहीं खोला जा सकता या केवल रीड-ओनली रखा जाता है। इस तरह, यदि एक वायरस के माध्यम से डाटा खो दिया है, कोई भी बैकअप का उपयोग करके फिर से शुरू कर सकता है। *(*जो हाल ही में अद्यतन किया गया होना चाहिए.*)* यदि ऑप्टिकल मीडिया *(optical media)* जैसे सीडी *(CD)* और डीवीडी *(DVD)* पर एक बैकअप सत्र बंद हो गया है, यह ***read-only*** बन जाता है और अब वायरस से संक्रमित नहीं हो सकता है। इसी तरह, बूट योग्य *(bootable)* पर एक ऑपरेटिंग सिस्टम का उपयोग कंप्यूटर को शुरू करने के लिए किया जा सकता है, यदि स्थापित आपरेटिंग सिस्टम उपयोग हीन हो गया है। अन्य विधि है भिन्न संचिका तंत्रों पर भिन्न आपरेटिंग सिस्टम का उपयोग.एक वायरस की दोनों को प्रभावित करने की सम्भावना नहीं होती है। डेटा बैकअप भी भिन्न संचिका तंत्रों पर डाला जा सकता है। उदाहरण के लिए, लिनक्स को ***NTFS*** *(NTFS)* विभाजन पर लिखने के लिए विशेष सॉफ़्टवेयर की आवश्यकता होती है, इसलिए यदि कोई सॉफ्टवेयर को इंस्टाल नहीं करता है और एक ***NTFS*** विभाजन पर एक बैकअप करने के लिए ***MS*** विन्डोज़ के अलग इंस्टालेशन का उपयोग करता है, बैकअप को लाइनक्स के किसी भी वर्जन से सुरक्षित होना चाहिए.इसी तरह, ***MS*** विन्डोज़ संचिका सिस्टम जैसे ***ext3*** *(ext3)*, को नहीं पढ़ सकता है, इसलिए यदि कोई सामान्यतया ***MS*** विन्डोज़ का उपयोग करता है, एक लिनक्स इंस्टालेशन का उपयोग करते हुए एक ***ext3*** विभाजन पर बैकअप बनाया जा सकता है।

पुनर्प्राप्ति विधियां

एक बार एक कंप्यूटर जब वायरस से समझौता कर लेता है, तो सामान्यतया ऑपरेटिंग सिस्टम को फ़िर से पूरी तरह से इंस्टाल किए बिना समान कंप्यूटर का उपयोग जारी रखना असुरक्षित होता है। लेकिन, कई विकल्प हैं जो कंप्यूटर में वायरस के आने के बाद उपस्थित होते हैं। ये क्रियाएँ वायरस के प्रकार की गंभीरता पर निर्भर करती हैं।

वायरस को हटाना

Windows Me *(Windows Me)*, ***Windows XP*** ***and*** ***Windows Vista*** पर एक सम्भावना है एक औजार जो सिस्टम रिस्टोर *(System Restore)* के नाम से जाना जाता है जो कि पिछले चेक बिन्दु के लिए रजिस्ट्री और जटिल तंत्र संचिका को रिस्टोर कर के रखता है। अक्सर एक वायरस एक सिस्टम को हेंग कर देता है और एक इसी के बाद से फ़िर से बूट होने पर सिस्टम के रिस्टोर बिन्दु में उसी दिन से समस्या आएगी.पिछले दिनों से रिस्टोर बिन्दु को दिया गया काम करना होता है, अब वायरस संग्रहीत संचिकाओं को संक्रमित नहीं कर पाता है। फिर भी कुछ वायरस, सिस्टम के रिस्टोर को और अन्य महत्वपूर्ण औजारों जैसे टास्क मेनेजर और कमांड प्रोम्प्ट को निष्क्रिय कर देते हैं। एक वायरस जो ऐसा करता है उसका उदहारण है ***CiaDoor***. प्रशासकोण के पास भिन्न कारणों के लिए सीमित उपयोगकर्ताओं से ऐसे औजारों को निष्क्रिय करने के विकल्प उपलब्ध हैं। वायरस ऐसा करने के लिए रजिस्ट्री में संशोधन करता है, केवल उस समय ऐसा नहीं होता है जब प्रशासक कंप्यूटर को नियंत्रित कर रहा हो, यह *सब* उपयोगकर्ताओं को औजारों को एक्सेस करने से रोकता है। जब एक संक्रमित औजार सक्रिय होता है तो यह एक संदेश देता है "आपके प्रशासक के द्वारा टास्क मेनेजर को अक्षम कर दिया गया है।"यहाँ तक की यदि उपयोगकर्ता जो प्रोग्राम खोलने की कोशिश कर रहा है वह प्रशासक है। उपयोगकर्ता जो माइक्रोसॉफ्ट ऑपरेटिंग सिस्टम को चला रहा है, एक मुक्त स्केन चलाने के लिए माइक्रोसॉफ्ट की वेबसाइट पर जा सकता है, यदि उनके पास उनका २० अंकों की पंजीकरण संख्या है।

ऑपरेटिंग सिस्टम की पुर्नस्थापना

ऑपरेटिंग सिस्टम को पुनः इंस्टाल करना वायरस हटाने का एक अलग तरीका है। इसमें साधारण रूप से *OS* विभाजन की पुनः फोर्मेटिंग और इसके मूल माध्यम से *OS* को इंस्टाल करना शामिल है। या एक साफ़ बेकअप छवि के साथ विभाजन की कल्पना *(imaging)* की जा सकती है *(*उदाहरण के लिए *Ghost (Ghost)* या *Acronis (Acronis)* के साथ लिया गया*)* . इस विधि का लाभ यह है की यह करने में सरल है, कई वायरस विरोधी स्कैन को चलाने की तुलना में यह तेज होता है और किसी भी मैलवेयर को दूर करने की गारंटी देता है।***Downsides*** में अन्य सभी सॉफ्टवेयर और ऑपरेटिंग सिस्टम का पुनः इंस्टाल करना शामिल है। उपयोगकर्ता डेटा को एक लाइव सीडी *(Live CD)* को बूट करके या हार्ड ड्राइव को किसी दूसरे कंप्यूटर में रखकर और उस कंप्यूटर के ऑपरेटिंग सिस्टम से बूट करके बेक अप किया जा सकता है। व्ब्फ्ज्ग्

यह भी देखिए

- एड वेयर *(Adware)*
- एंटीवायरस सॉफ्टवेयर *(Antivirus software)*
- अटेक ट्री *(Attack tree)*
- ब्लेक हेट *(Black hat)*
- कम्प्यूटर असुरक्षा *(Computer insecurity)*
- कम्प्यूटर का कीड़ा *(Computer worm)*
- *Crimeware (Crimeware)*
- क्रिप्टोवाईरोलोजी *(Cryptovirology)*
- किस्ट्रोक लॉगिंग *(Keystroke logging)*
- कंप्यूटर वायरस की सूची *(List of computer viruses)*
- कंप्यूटर वायरस *hoaxes* की सूची *(List of computer virus hoaxes)*
- लाईनक्स मालवेयर *(Linux malware)*
- मालवेयर/मैलवेयर *(Malware)*
- गतिशील वाईरस *(Mobile virus)*
- अनेक भागों में विभक्त वाईरस *(Multipartite virus)*
- पाम *OS* वाइरस *(Palm OS Viruses)*
- *obscurity* के माध्यम से सुरक्षा *(Security through obscurity)*
- स्पैम *(Spam)*
- स्पायवेयर *(Spyware)*

- उल्लेखनीय कंप्यूटर वायरस और वर्म की समयरेखा *(Timeline of notable computer viruses and worms)*
- *Trojan* हार्स *(*कम्प्यूटिंग*) (Trojan horse (computing))*
- वायरस *hoax (Virus hoax)*
- एइकार परीक्षण संचिका *(Eicar test file)*

== नोट्स*A computer virous a type of electronce code that is used in computer there is a hoary to end tha information in acomputer program a telephone line can be suppressed by some means this code get a wrong information information collect Ed by tha creater can be and if a computer is connected to a netweb because of being connected to a electronics*

सन्दर्भ

- *Mark Russinovich, Advanced Malware Cleaning* ***video*** *(https://web.archive.org/web/20070630223429/http://www.microsoft.com/emea/itsshowtime/sessionh.aspx?videoid=359) ,Microsoft TechEd: IT Forum,* नवम्बर २००६

बाहरी कड़ियाँ

- वायरस हुआ *25* साल का और कारोबार *16* अरब अमेरिकी डालर का *(https://web.archive.org/web/20110203090425/http://hindi.economictimes.indiatimes.com/articleshow/7331955.cms)*
- अमेरिकी सरकार *CERT (*कंप्यूटर आपातकालीन तैयारी दल*)* साइट *(https://web.archive.org/web/20081111045032/http://www.us-cert.gov/)*

अन्य टेक्स्ट

- वीडियो : *"*सूचना प्रौद्योगिकी की सुरक्षा के बारे में माइक्रोसॉफ्ट सम्मेलन-- मांग पर वीडियो *(http://www.microsoft.com/emea/itsshowtime/result_search.aspx?track=1&x=37&y=7) "*
- अनुच्छेद : *"'* कंप्यूटर वायरस क्या है *(https://www.rexgin.in/2020/07/computer-virus-in-hindi.html) "*
- अनुच्छेद : *"*कंप्यूटर वायरस कैसे काम करता है *(https://web.archive.org/web/20130629114608/http://www.howstuffworks.com/virus.htm) "*
- अनुच्छेद : *"*पीसी वाइरस का संक्षिप्त इतिहास *(https://web.archive.org/web/20081103101042/http://vx.netlux.org/lib/aas14.html) " (*प्रारंभिक*)* डॉ॰ अनिल सुलैमान के द्वारा

- अनुच्छेद : "क्या ' अच्छे ' कंप्यूटर वायरस अभी भी एक खराब विचार है ? (https://web.archive.org/web/20081204060902/http://vx.netlux.org/lib/avb02.html) "
- अनुच्छेद : "आपके ईमेल की वाइरस और अन्य मैलवेयर सुरक्षा (https://web.archive.org/web/20081112071616/http://www.windowsecurity.com/articles/Protecting_Email_Viruses_Malware.html) "
- अनुच्छेद : "हैकिंग Counterculture पर दूर (https://web.archive.org/web/20090122174144/http://www.iath.virginia.edu/pmc/text-only/issue.990/ross-1.990) " एंड्रयू रॉस (Andrew Ross) के द्वारा.
- अनुच्छेद : "अंतरिक्ष में जानकारी के क्षेत्र में एक वायरस (https://web.archive.org/web/20081205043220/http://journal.media-culture.org.au/0406/07_Sampson.php) " टोनी Sampson के द्वारा
- अनुच्छेद : "डॉ॰ Aycock के खराब विचार (https://web.archive.org/web/20081205030533/http://journal.media-culture.org.au/0502/02-sampson.php) " टोनी Sampson के द्वारा
- अनुच्छेद : "Digital Monsters, Binary Aliens (https://web.archive.org/web/20081203173214/http://journal.fibreculture.org/issue4/issue4_parikka.html) " जूसी परिका (Jussi Parikka) के द्वारा
- अनुच्छेद : "The Universal Viral Machine (https://web.archive.org/web/20081002005741/http://www.ctheory.net/articles.aspx?id=500) " जूसी परिका (Jussi Parikka) के द्वारा
- अनुच्छेद : "Hypervirus: A Clinical Report (https://web.archive.org/web/20081002005756/http://www.ctheory.net/articles.aspx?id=504) " थियरी बर्दिनी (Thierry Bardini) के द्वारा.
- अनुच्छेद : "The Cross-site Scripting Virus (https://web.archive.org/web/20140823161243/http://www.bindshell.net/papers/xssv/) "
- आरएफसी ११३५ इंटरनेट के Helminthiasis
- अनुच्छेद: "The Virus Underground (https://web.archive.org/web/20090122061129/http://www.cse.msu.edu/~cse825/virusWriter.htm) "

1. सबसे पहला वायरस (https://kyahai.in/malware-virus-kya-hai-computer-virus-detail-in-hindi-2019/)
2.
3. "बूट क्षेत्र के वायरस की मरम्मत" (https://web.archive.org/web/20110112024842/http://antivirus.about.com/od/securitytips/a/bootsectorvirus.htm) . मूल (http://antivirus.about.com/od/securitytips/a/bootsectorvirus.htm) से 12 जनवरी 2011 को पुरालेखित. अभिगमन तिथि 13 नवंबर 2008.
4. डॉ॰ सुलैमान का वायरस विश्वकोश, १९९५, ISBN १-८९७६६१-००-२, पर अवशोषितhttp://vx.netlux.org/lib/aas10.html Archived (https://web.archive.org/web/20120117091338/http://vx.netlux.org/lib/aas10.html) 2012-01-17 at the Wayback Machine

5.
6.
7.
8. "एक कंप्यूटर वायरस की आवश्यकता है ?-- अब डाउनलोड करें" (https://web.archive.org/web/20081020114307/http://www.infoniac.com/offbeat-news/computervirus.html) . मूल (http://www.infoniac.com/offbeat-news/computervirus.html) से 20 अक्तूबर 2008 को पुरालेखित. अभिगमन तिथि 13 नवंबर 2008.
9.
10. "एप्पल -- एक मैक प्राप्त करें" (http://www.apple.com/getamac) . मूल से 2 दिसंबर 2008 को पुरालेखित (https://web.archive.org/web/20081202085408/http://www.apple.com/getamac/) . अभिगमन तिथि 13 नवंबर 2008.
11.
12.

Last edited 2 months ago by 2405:204:A312:25E9:95D6:5EC7:AF95:6A20

विकिपीडिया