

CHƯƠNG 12 KIỂM SOÁT MẠNG

ThS. Trần Bá Nhiệm
Website:
sites.google.com/site/tranbanhiem
Email: tranbanhiem@gmail.com

Nội dung

- Giới thiệu
- DNS
- Ping
- WHOIS
- Một số nội dung khác của TCP/IP
- WMI

Giới thiệu

- Các giao thức mạng không chỉ dùng để truyền dữ liệu từ điểm này sang điểm khác
- Một số giao thức có mục đích đặc biệt giúp quản lý lưu thông dữ liệu Internet và làm cho việc dùng mạng trở nên dễ dàng hơn
- Nội dung của chương bàn về 4 giao thức DNS, WHOIS, Ping và WMI

29/06/2011

Chương 12: Kiểm soát mạng

3

DNS

- DNS dùng UDP port 53 và được mô tả ở RFC 1010, RFC 1304, RFC 1035, RFC 1183.
- DNS là giao thức dùng để phân giải tên miền thành địa chỉ IP
- Không có máy tính trung tâm nào lưu trữ danh sách domain cùng với địa chỉ IP mà một mạng toàn cầu các DNS làm nhiệm vụ này

29/06/2011

Chương 12: Kiểm soát mạng

4

DNS

- Các DNS server đều đặn trao đổi, cập nhật thông tin và làm cho thông tin lan truyền trên mạng Internet
- Nếu chúng ta cập thay đổi hosting, thì cần 48 giờ để thông tin DNS mới có thể cập nhật được
- Chúng ta có thể dùng `DNS.GetHostByName` để chuyển tên miền (string) thành một địa chỉ IP (`IPHostEntry`)

29/06/2011

Chương 12: Kiểm soát mạng

5

DNS

- Tiến trình ngược lại, chuyển một địa chỉ IP thành một tên miền có thể dùng `DNS.GetHostByAddress`
- Một khía cạnh thú vị của DNS là vai trò gửi, nhận email đã bàn trong chương 5. SMTP server tìm địa chỉ POP3 server đích bằng cách dùng DNS mail exchange (MX)
- Các MX trả về một danh sách các mail server liên kết với một domain nào đó

29/06/2011

Chương 12: Kiểm soát mạng

6

DNS

- Kỹ thuật trên có thể được dùng để kiểm tra địa chỉ email hợp lệ
- Cũng được dùng để đơn giản hóa địa chỉ email, bỏ qua việc nhập phần chi tiết của SMTP server
- Thuận lợi cuối cùng là giúp SMTP chuyển email nhanh hơn, cải thiện hiệu suất

29/06/2011

Chương 12: Kiểm soát mạng

7

Hiện thực DNS MX

- Tạo project mới, 1 form, 3 textbox tên tbDNSServer, tbDomain, tbStatus, 1 button tên btnFind
- Xử lý sự kiện Click của nút lệnh trên:

```
private void btnFind_Click(object sender, EventArgs e)
{
    byte[] DNSQuery;
    byte[] DNSReply;
    UdpClient dnsClient = new
    UdpClient(tbDNSServer.Text, 53);
```

29/06/2011

Chương 12: Kiểm soát mạng

8

Hiện thực DNS MX

```

DNSQuery =
makeQuery(DateTime.Now.Millisecond * 60,
tbDomain.Text);
    dnsClient.Send(DNSQuery,
DNSQuery.GetLength(0));
    IPEndPoint endpoint = null;
    DNSReply = dnsClient.Receive(ref endpoint);
    this.tbStatus.Text =
makeResponse(DNSReply, tbDomain.Text);
}

```

29/06/2011

Chương 12: Kiểm soát mạng

9

Hiện thực DNS MX

- Đoạn code trên mở kết nối UDP port 53 đến DNS server và gửi một MX query đến nó. Response nhận được sẽ chuyển cho hàm makeResponse(...)
- Để chuẩn bị MX query chúng ta phải viết một hàm mới makeQuery(...) – chi tiết giải thích công việc của hàm này tham khảo thêm trong các RFC

29/06/2011

Chương 12: Kiểm soát mạng

10

Hiện thực DNS MX

```
public byte[] makeQuery(int id, string name)
{
    byte[] data = new byte[512];
    byte[] Query;
    data[0] = (byte)(id >> 8);
    data[1] = (byte)(id & 0xFF);
    data[2] = (byte)1; data[3] = (byte)0;
    data[4] = (byte)0; data[5] = (byte)1;
    data[6] = (byte)0; data[7] = (byte)0;
    data[8] = (byte)0; data[9] = (byte)0;
    data[10] = (byte)0; data[11] = (byte)0;
```

29/06/2011

Chương 12: Kiểm soát mạng

11

Hiện thực DNS MX

```
string[] tokens = name.Split(new char[] { '.' });
string label;
int position = 12;
for (int j = 0; j < tokens.Length; j++){
    label = tokens[j];
    data[position++] = (byte)(label.Length & 0xFF);
    byte[] b =
System.Text.Encoding.ASCII.GetBytes(label);
    for (int k = 0; k < b.Length; k++)
    {
        data[position++] = b[k];
    }
}
```

29/06/2011

Chương 12: Kiểm soát mạng

12

Hiện thực DNS MX

```

    }
    data[position++] = (byte)0; data[position++] = (byte)0;
    data[position++] = (byte)15; data[position++] = (byte)0;
    data[position++] = (byte)1;
    Query = new byte[position + 1];
    for (int i = 0; i <= position; i++){
        Query[i] = data[i];
    }
    return Query;
}

```

29/06/2011

Chương 12: Kiểm soát mạng

13

Hiện thực DNS MX

- Các tên domain trong DNS query có định dạng bất thường, thay vì dấu chấm phân cách mỗi level thì lại dùng 1 byte thể hiện phần kế tiếp. Ví dụ: www.google.com trở thành 3www6google3com
- Bước kế tiếp là phân tích response, do đó cần tạo hàm makeResponse như sau:

29/06/2011

Chương 12: Kiểm soát mạng

14

Hiện thực DNS MX

```
public string makeResponse(byte[] data, string name)
{
    int qCount = ((data[4] & 0xFF) << 8) | (data[5] & 0xFF);
    int aCount = ((data[6] & 0xFF) << 8) | (data[7] & 0xFF);
    int position = 12;
    for (int i = 0; i < qCount; ++i){
        name = "";
        position = proc(position, data, ref name);
        position += 4;
    }
}
```

29/06/2011

Chương 12: Kiểm soát mạng

15

Hiện thực DNS MX

```
string Response = "";
for (int i = 0; i < aCount; ++i){
    name = "";
    position = proc(position, data, ref name);
    position += 12;
    name = "";
    position = proc(position, data, ref name);
    Response += name + "\r\n";
}
return Response;
}
```

29/06/2011

Chương 12: Kiểm soát mạng

16

Hiện thực DNS MX

- Đoạn chương trình trên thực hiện khai thác MX server từ DNS reply và hiển thị chúng lên màn hình.
- Sau đó thực hiện hàm proc để chuyển đổi giữa định dạng DNS tự nhiên cho các tên domain và định dạng chuẩn dấu “.”
- Mã chương trình cho proc như sau:

29/06/2011

Chương 12: Kiểm soát mạng

17

Hiện thực DNS MX

```
private int proc(int position, byte[] data, ref string
name)
{
    int len = (data[position++] & 0xFF);
    if (len == 0)
        return position;
    int offset;
    do {
        if ((len & 0xC0) == 0xC0){
            if (position >= data.GetLength(0))
                return -1;

```

29/06/2011

Chương 12: Kiểm soát mạng

18

Hiện thực DNS MX

```

offset = ((len & 0x3F) << 8) | (data[position++] &
0xFF);
proc(offset, data, ref name);
return position;
}
else {
if ((position + len) > data.GetLength(0))
return -1;
name += Encoding.ASCII.GetString(data,
position, len);
position += len;
}

```

29/06/2011

Chương 12: Kiểm soát mạng

19

Hiện thực DNS MX

```

if (position > data.GetLength(0))
return -1;
len = data[position++] & 0xFF;
if (len != 0)
name += ".";
} while (len != 0);
return position;
}

```

29/06/2011

Chương 12: Kiểm soát mạng

20

Hiện thực DNS MX

- Để chuẩn bị thực hiện thí nghiệm trên, cần tìm địa chỉ IP của một DNS server. Chúng ta có thể dùng địa chỉ 204.111.1.36 hoặc tùy theo ISP của mạng
- Nhập tên domain, không cần phần www đứng trước vào textbox thứ 2

29/06/2011

Chương 12: Kiểm soát mạng

21

Hiện thực DNS MX

The image shows two side-by-side screenshots of a Windows application window titled 'Form1'. Each window contains a 'DNS Server' field with the value '203.113.188.2', a 'Domain' field, a 'Find' button, and a 'DNS MX's list:' label above a text area.

Left Screenshot (Domain: hotmail.com):

DNS Server: 203.113.188.2
Domain: hotmail.com
Find
DNS MX's list:
mx3.hotmail.com
mx4.hotmail.com
mx1.hotmail.com
mx2.hotmail.com

Right Screenshot (Domain: gmail.com):

DNS Server: 203.113.188.2
Domain: gmail.com
Find
DNS MX's list:
alt1.gmail-smtp-in1.google.com
alt3.gmail-smtp-in1.google.com
gmail-smtp-in1.google.com
alt2.gmail-smtp-in1.google.com
alt4.gmail-smtp-in1.google.com

29/06/2011

Chương 12: Kiểm soát mạng

22

Ping

- Ping hay Internet control message protocol (ICMP), là giao thức dùng để thông báo các kết nối bị ngắt, các vấn đề về mức router mà client có thể chạm đến
- Khi một router không thể lấy gói tin từ hop kế, nó hủy gói và gửi gói ICMP về cho server

29/06/2011

Chương 12: Kiểm soát mạng

23

Ping

- Các gói ICMP không dùng để thông báo về các vấn đề routing hòng cho các gói tin ICMP khác để tránh hiệu ứng thác đổ
- Nhiều người quen thuộc với ứng dụng ping để kiểm tra một máy tính có hoạt động trên mạng và độ trễ khi kết nối đến nó
- Chúng ta sẽ chọn cách hiện thực ping đơn giản dùng ICMP DLL

29/06/2011

Chương 12: Kiểm soát mạng

24

Hiện thực Ping

- Tạo project mới, thêm class PING như sau:

29/06/2011

Chương 12: Kiểm soát mạng

25

Hiện thực Ping

- `IcmpCreateFile` tạo một trình quản lý tài nguyên để dùng khi sinh ra ping request
- Khi một chương trình phát ra một số lượng lớn ping request thì `IcmpCloseHandle` được dùng để yêu cầu bộ nhớ
- `IcmpSendEcho` gửi một ICMP echo request đến host xác định trong tham số `DestAddress`

29/06/2011

Chương 12: Kiểm soát mạng

26

Hiện thực Ping

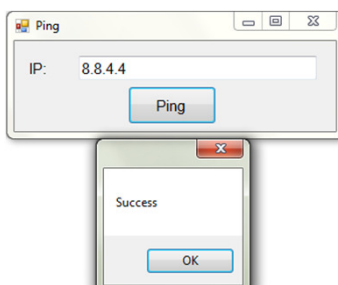
- Dạng thức của gói tin ping đi ra được thiết lập trong tham số RequestOptions
- Chi tiết của reply (nếu có) được lưu trữ trong ReplyBuffer
- Trở lại project, thêm 1 form, 1 textbox tên tbIP và 1 button tên btnPing.
- Xử lý sự kiện Click của button như sau:

29/06/2011

Chương 12: Kiểm soát mạng

27

Hiện thực Ping



29/06/2011

Chương 12: Kiểm soát mạng

28

WHOIS

- WHOIS là một giao thức có thể dùng để truy vấn tên domain.
- WHOIS chạy trên TCP port 43 và được mô tả trong RFC 954
- Thông tin bao gồm tên, công ty mua domain cùng với chi tiết về DNS server, cũng như người điều hành domain đó

29/06/2011

Chương 12: Kiểm soát mạng

29

WHOIS

- Hầu hết các quốc gia đều có WHOIS server riêng quản lý tên top-level domain như .co,.uk,.ie, .vn
- Các tên top-level domain quốc tế như .com, .net, .org được lưu trữ trong các server của tập hợp WHOIS server rất lớn. Một vài trong đó nổi tiếng như:
whois.networksolutions.com,
whois.crsnic.net, whois.ripe.net

29/06/2011

Chương 12: Kiểm soát mạng

30

WHOIS

- Để thực hiện một WHOIS query thủ công, chạy telnet và gõ vào dòng lệnh, ví dụ:
O whois.ripe.net 43
Google.de
- Thực hiện WHOIS query với .NET tương đối dễ dàng. Tất cả công việc cần làm chỉ là mở một kết nối TCP port 43, gửi tên domain, đọc response cho đến khi kết nối đóng

29/06/2011

Chương 12: Kiểm soát mạng

31

Hiện thực WHOIS

- Tạo project, gồm 1 form, 3 textbox tên tbServer, tbQuery, tbStatus, 1 button tên btnSend.
- Thiết lập thuộc tính multiline của tbStatus bằng true
- Xử lý sự kiện Click của button:

29/06/2011

Chương 12: Kiểm soát mạng

32

Hiện thực WHOIS

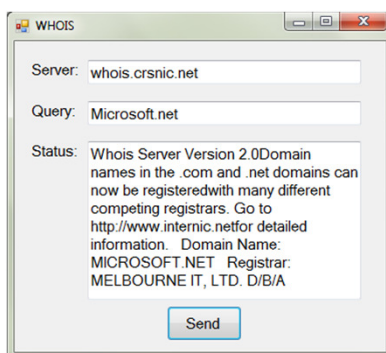
```
private void btnSend_Click(object sender, EventArgs e)
{
    byte[] Query = Encoding.ASCII.GetBytes(tbQuery.Text +
    "\n");
    TcpClient clientSocket = new TcpClient(tbServer.Text, 43);
    NetworkStream networkStream =
    clientSocket.GetStream();
    networkStream.Write(Query, 0, Query.GetLength(0));
    StreamReader Response = new
    StreamReader(networkStream);
    tbStatus.Text = Response.ReadToEnd();
    networkStream.Close();
}
```

29/06/2011

Chương 12: Kiểm soát mạng

33

Hiện thực WHOIS



29/06/2011

Chương 12: Kiểm soát mạng

34

Telnet

- Trước khi có GUI, những người dùng UNIX vẫn thích thú với việc giao tiếp bằng dòng lệnh
- Cho đến bây giờ vẫn còn nhiều host online được cài đặt UNIX hoặc Linux nên telnet vẫn là chuẩn thực tế của một số server chạy trên hệ điều hành đó
- telnet chạy trên TCP port 23

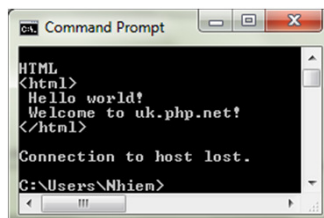
29/06/2011

Chương 12: Kiểm soát mạng

35

Telnet

- Nếu một webserver được cài đặt trên máy tính, chúng ta có thể kiểm tra telnet bằng cách nhập lần lượt vào các lệnh sau:
- telnet localhost 80
- GET /



29/06/2011

Chương 12: Kiểm soát mạng

36

Một số thành phần khác trong bộ TCP/IP

- Có rất nhiều giao thức làm việc phía sau IP network để cung cấp dịch vụ.
- Thông thường không dùng chúng để lập trình, nhưng chúng cũng xứng đáng để nghiên cứu:
 - ARP
 - RARP
 - RIP
 - OSPF
 - BGP/EGP
 - SNMP
 - PPP
 - WMI

29/06/2011

Chương 12: Kiểm soát mạng

37

Một số thành phần khác trong bộ TCP/IP

- ARP là viết tắt của Address resolution protocol phân giải địa chỉ IP sang MAC. Reverse ARP (RARP) làm tiến trình ngược lại
- Thao tác dòng lệnh:
arp -a

29/06/2011

Chương 12: Kiểm soát mạng

38

Một số thành phần khác trong bộ TCP/IP

- Routing information protocol (RIP) làm việc bằng cách đếm số lần gói di chuyển về đích. Mỗi routing mới được gọi là 1 hop. Số hop tối đa được thiết lập là 16. RIP sẽ hủy những gói có số hop trên 16
- Open shortest path first (OSPF) là giao thức routing dùng giải thuật link-state. OSPF không có giới hạn số hop tối đa.

29/06/2011

Chương 12: Kiểm soát mạng

39

Một số thành phần khác trong bộ TCP/IP

- Border gateway protocol (BGP) thay thế exterior gateway protocol (EGP) và được dùng để gửi các gói ra ngoài đến mạng khác. BGP khác OSPF ở chỗ chỉ sử dụng trong mạng nội bộ. Chú ý: không bao giờ có 2 BGP router trên cùng mạng mà không có hỗ trợ của OSPF hoặc RIP

29/06/2011

Chương 12: Kiểm soát mạng

40

Một số thành phần khác trong bộ TCP/IP

- Simple network management protocol (SNMP) cho phép những người quản trị mạng kết nối và quản lý các thiết bị mạng. Nó đang bị thay thế bởi RMON nhưng vẫn còn sử dụng phổ biến. SNMP hoạt động trên UDP port 161. SNMP có thể báo các sự kiện lỗi đối với các thiết bị mạng

29/06/2011

Chương 12: Kiểm soát mạng

41

Một số thành phần khác trong bộ TCP/IP

- Point-to-point protocol (PPP) dùng để vận chuyển IP, IPX, NetBEUI trên các kết nối nối tiếp như modem. PPP thường dùng bởi các ISP để cung cấp thuê bao truy cập Internet bằng modem hoặc ISDN. PPP yêu cầu một số điện thoại và thông thường có địa chỉ DNS server cùng với username, password. PPP thay thế Serial Line Internet Protocol (SLIP) vì tốc độ, tính đơn giản và linh hoạt

29/06/2011

Chương 12: Kiểm soát mạng

42

Một số thành phần khác trong bộ TCP/IP

- WMI (Windows Management Instrumentation) được dùng trong Windows intranet để cung cấp các tác vụ quản trị đơn giản từ xa. Thuận lợi chính là WMI client đã được tích hợp sẵn trong Windows, nên không cần cài đặt client
- Một trong những ứng dụng chính của WMI là khai thác thông tin về hệ điều hành Windows từ xa

29/06/2011

Chương 12: Kiểm soát mạng

43

Một số thành phần khác trong bộ TCP/IP

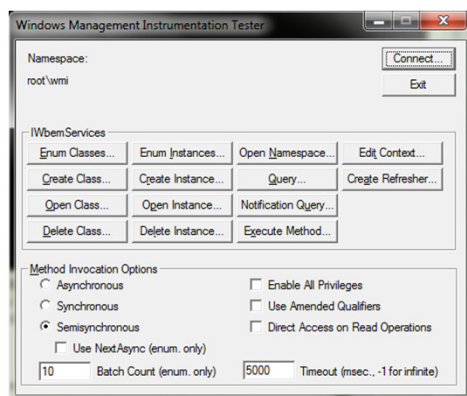
- WMI có cấu trúc giống một cơ sở dữ liệu. CIM (Common Information Model) chứa nhiều namespace, trong đó có nhiều class với các thuộc tính tương ứng các thiết bị như CD-ROM, NT event log,...
- Để xem CIM, chạy WBEMTEST từ dấu nhắc lệnh, chọn Connect→gõ Root→Connect→...

29/06/2011

Chương 12: Kiểm soát mạng

44

Một số thành phần khác trong bộ TCP/IP



29/06/2011

Chương 12: Kiểm soát mạng

45

Một số thành phần khác trong bộ TCP/IP

- Một số namespace quan trọng như:
 - root\directory\ldap: cung cấp truy xuất vào các dịch vụ active directory
 - root\snmp: cung cấp truy xuất vào cơ sở dữ liệu SNMP MIB
 - root\default: cung cấp truy xuất vào registry
 - root\WMI: cung cấp truy xuất vào các thiết bị Windows Device Model (WDM)

29/06/2011

Chương 12: Kiểm soát mạng

46

Đọc dữ liệu WMI

- Dữ liệu WMI giống khái niệm cơ sở dữ liệu, nhưng namespace System.Management dùng để đóng gói WMI thì không
- Tương tự muốn thực hiện lệnh SQL thì phải thực hiện kết nối, một phạm vi phải được định nghĩa trước khi muốn dùng WQL

29/06/2011

Chương 12: Kiểm soát mạng

47

Đọc dữ liệu WMI

- WMI dùng ManagementScope để truyền vị trí của máy tính từ xa theo dạng \\<host name>\root\namespace và một đối tượng ConnectionOptions để chứa chứng chỉ đăng nhập (username, password).
- ManagementObjectSearcher xử lý WQL. Khi gọi Get() thì đối tượng này trả về một ManagementObjectCollection

29/06/2011

Chương 12: Kiểm soát mạng

48

Đọc dữ liệu WMI

- ManagementObjectCollection tương tự table, trong đó mỗi phần tử là một dòng. Dòng này được thể hiện bằng đối tượng ManagementBaseObject. Mỗi dòng có một số lượng cột thay đổi, được thể hiện bởi đối tượng PropertyData, chúng giữ bên trong một Properties collection chứa đối tượng ManagementBaseObject

29/06/2011

Chương 12: Kiểm soát mạng

49

Hiện thực đọc dữ liệu WMI

- Tạo project mới, gồm 1 form, 4 textbox tên tbHost, tbUsername, tbPassword, tbExecute; 1 list view tên lvWMI và 1 button tên btnExecute.
- Thêm tham chiếu vào project: chọn Project→Add References và trở đến System.Management
- Xử lý sự kiện Click của button:

29/06/2011

Chương 12: Kiểm soát mạng

50

Hiện thực đọc dữ liệu WMI

```
private void btnExecute_Click(object sender, EventArgs e)
{
    ConnectionOptions Options = new ConnectionOptions();
    if (tbPassword.Text != "" && tbUsername.Text != "")
    {
        Options.Username = tbHost.Text + "\\" + tbUsername.Text;
        Options.Password = tbPassword.Text;
    }
    ManagementScope Scope = new ManagementScope("\\\\" +
    tbHost.Text +
    "\\root\\cimv2", Options);
    Scope.Connect();
}
```

29/06/2011

Chương 12: Kiểm soát mạng

51

Hiện thực đọc dữ liệu WMI

```
ObjectQuery Query = new
ObjectQuery(tbExecute.Text);
ManagementObjectSearcher Searcher = new
ManagementObjectSearcher(Scope, Query);
ManagementObjectCollection ItemCollection;
ItemCollection = Searcher.Get();
lvWMI.Clear();
lvWMI.Columns.Clear();
lvWMI.View = View.Details;
foreach (ManagementBaseObject Item in
ItemCollection) {
}
```

29/06/2011

Chương 12: Kiểm soát mạng

52

Hiện thực đọc dữ liệu WMI

```

if (lvWMI.Columns.Count == 0) {
    foreach (PropertyData prop in Item.Properties)
    {
        lvWMI.Columns.Add(prop.Name, lvWMI.Width
/ 4, HorizontalAlignment.Left);
    }
}
ListViewItem lvItem = new ListViewItem();
bool firstColumn = true;
foreach (PropertyData prop in Item.Properties) {
    if (firstColumn)
    {

```

29/06/2011

Chương 12: Kiểm soát mạng

53

Hiện thực đọc dữ liệu WMI

```

lvItem.SubItems[0].Text = prop.Value + "";
    firstColumn = false;
}
else
{
    lvItem.SubItems.Add(prop.Value + "");
}
}
lvWMI.Items.Add(lvItem);
}
}

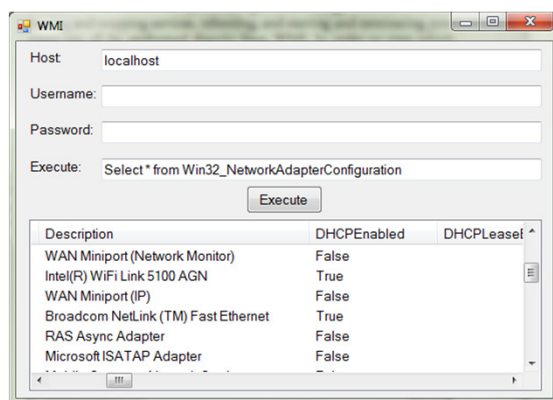
```

29/06/2011

Chương 12: Kiểm soát mạng

54

Hiện thực đọc dữ liệu WMI



29/06/2011

Chương 12: Kiểm soát mạng

55

Hiện thực đọc dữ liệu WMI

- Chú ý: khai báo thêm namespace using System.Management;
- Để truy vấn WMI tới máy tính ở xa, chúng ta cần cung cấp thêm username và password có quyền administrator của máy tính đó

29/06/2011

Chương 12: Kiểm soát mạng

56

Leveraging WMI

- Nếu chúng ta không bị hạn chế quyền đọc dữ liệu khi dùng WMI thì cũng có thể thực hiện tác vụ này trên máy ở xa. Các chức năng như khởi động và dừng dịch vụ, khởi động lại, kết thúc các tiến trình có thể thực hiện trực tiếp từ WMI
- Để xem phương thức nào có thể gọi từ WMI class đã cho, chạy lệnh WBEMTEST

29/06/2011

Chương 12: Kiểm soát mạng

57

Leveraging WMI

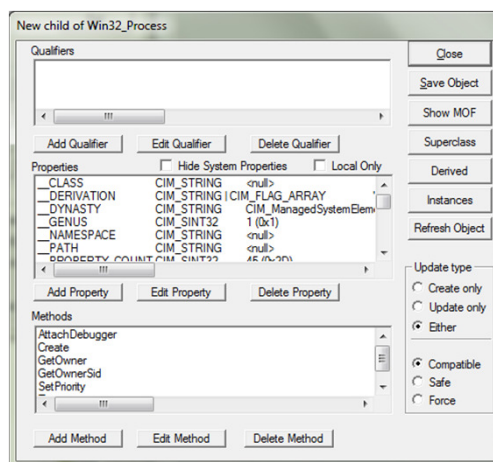
- Kết nối vào namespace (ví dụ: root\cimv2), click Create Class sau đó gõ vào tên class (ví dụ: WIN32_PROCESS).
- Các phương thức được hỗ trợ sẽ liệt kê lên màn hình

29/06/2011

Chương 12: Kiểm soát mạng

58

Leveraging WMI



29/06/2011

Chương 12: Kiểm soát mạng

59

Leveraging WMI

- Giống như ví dụ trước, một kết nối, một phạm vi được yêu cầu cho máy tính từ xa
- Thay cho việc thực thi WQL query, khởi tạo một ManagementClass chứa Win32_Process class. Class này có phương thức Create có thể sinh ra tiến trình mới. Phương thức truyền các tham số thông qua đối tượng ManagementBaseObject

29/06/2011

Chương 12: Kiểm soát mạng

60

Leveraging WMI

- Tạo project mới, gồm 1 form, 4 textbox tên tbHost, tbUsername, tbPassword, tbExecute; 1 list view tên lvWMI và 1 button tên btnExecute.
- Thêm tham chiếu vào project: chọn Project→Add References và trở đến System.Management
- Xử lý sự kiện Click của button:

29/06/2011

Chương 12: Kiểm soát mạng

61

Leveraging WMI

```
private void btnExecute_Click(object sender, EventArgs e)
{
    ConnectionOptions Options = new ConnectionOptions();
    if (tbPassword.Text != "" && tbUsername.Text != "") {
        Options.Username = tbHost.Text + "\\" +
        tbUsername.Text;
        Options.Password = tbPassword.Text;
    }
    ManagementScope Scope = new
    ManagementScope("\\\\" + tbHost.Text + "\\root\\cimv2",
    Options);
```

29/06/2011

Chương 12: Kiểm soát mạng

62

Leveraging WMI

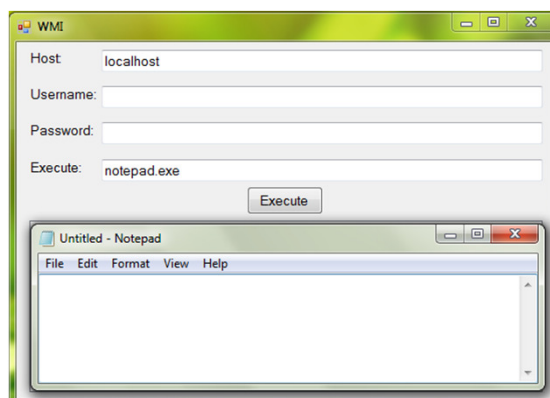
```
Scope.Connect();  
ManagementClass ProcessClass = new  
ManagementClass("Win32_Process");  
ManagementBaseObject inParams =  
ProcessClass.GetMethodParameters("Create");  
ProcessClass.Scope = Scope;  
inParams["CommandLine"] = tbExecute.Text;  
ProcessClass.InvokeMethod("Create", inParams,  
null);  
}
```

29/06/2011

Chương 12: Kiểm soát mạng

63

Leveraging WMI



29/06/2011

Chương 12: Kiểm soát mạng

64

Leveraging WMI

- Vẫn giống như ví dụ trước, chúng ta khai báo thêm namespace
using System.Management;
- Để thực hiện trên máy tính ở xa, chúng ta cần cung cấp thêm username và password có quyền administrator của máy tính đó

29/06/2011

Chương 12: Kiểm soát mạng

65

Bài tập

- Cài đặt các chương trình đã minh họa trong bài giảng của chương bằng ngôn ngữ C# hoặc VB.NET

29/06/2011

Chương 12: Kiểm soát mạng

66